



# Postgres Enterprise Manager

Version 8

1	Postgres Enterprise Manager	8
2	Release notes	8
2.1	Postgres Enterprise Manager 8.7.1 release notes	9
2.2	Postgres Enterprise Manager 8.7.0 release notes	9
2.3	Postgres Enterprise Manager 8.6.4 release notes	9
2.4	Version 8.6.3	9
2.5	Version 8.6.1	10
2.6	Version 8.6.0	10
2.7	Version 8.5.0	10
2.8	Version 8.4.0	11
2.9	Version 8.3.0	11
2.10	Version 8.2.0	12
2.11	Version 8.1.1	13
2.12	Version 8.1.0	14
2.13	Version 8.0.1	16
2.14	Version 8.0.0	17
3	Platform compatibility	19
4	Hardware requirements for installing the PEM server on Linux	19
5	Understanding Postgres Enterprise Manager components and architecture	19
6	Deployment considerations	23
6.1	Using Failover Manager for high availability	23
6.2	Connection pooling using pgBouncer	32
6.2.1	PEM server and agent connection management mechanism	33
6.2.2	Preparing the PEM database server	34
6.2.3	Configuring PgBouncer	35
6.2.4	Configuring the PEM agent	38
6.3	Authentication options	39
6.3.1	Configuring the PEM server to use Kerberos authentication	40
6.3.2	Configuring the PEM server to use Windows Active Directory domain services for Kerberos authentication (SSPI)	
6.3.3	Configuring the PEM server to use two-factor authentication	5144
6.4	Securing your deployment	52
6.4.1	Apache HTTPD security configurations	53
6.4.2	PEM application security configurations	58
6.5	Web server installation options	61
7	Installing Postgres Enterprise Manager server	61
7.1	Prerequisites to install the PEM server on Linux	62
7.2	Dependencies of the PEM Server and Agent on Linux	64
7.3	Installing Postgres Enterprise Manager server on Linux x86 (amd64)	68
7.3.1	Installing Postgres Enterprise Manager server on RHEL 9 or OL 9 x86_64	69
7.3.2	Installing Postgres Enterprise Manager server on RHEL 8 or OL 8 x86_64	71
7.3.3	Installing Postgres Enterprise Manager server on AlmaLinux 9 or Rocky	73
7.3.4	Installing Postgres Enterprise Manager server on AlmaLinux 8 or Rocky	74
7.3.5	Installing Postgres Enterprise Manager server on SLES 15 x86_64	76
7.3.6	Installing Postgres Enterprise Manager server on SLES 12 x86_64	78
7.3.7	Installing Postgres Enterprise Manager server on Ubuntu 22.04 x86_64	80
7.3.8	Installing Postgres Enterprise Manager server on Ubuntu 20.04 x86_64	82
7.3.9	Installing Postgres Enterprise Manager server on Debian 11 x86_64	84
7.3.10	Installing Postgres Enterprise Manager server on CentOS 7 x86_64	86
7.3.11	Installing Postgres Enterprise Manager server on Debian 10 x86_64	87
7.3.12	Installing Postgres Enterprise Manager server on RHEL 7 or OL 7 x86_64	89
7.4	Installing Postgres Enterprise Manager server on Linux IBM Power (ppc64le)	91
7.4.1	Installing Postgres Enterprise Manager server on RHEL 9 ppc64le	92
7.4.2	Installing Postgres Enterprise Manager server on RHEL 8 ppc64le	93
7.4.3	Installing Postgres Enterprise Manager server on SLES 15 ppc64le	95
7.4.4	Installing Postgres Enterprise Manager server on SLES 12 ppc64le	97
7.5	Installing Postgres Enterprise Manager server on Windows	99



7.5.1	Installing the PEM server and PEM-HTTPD on the same host	100
7.5.2	Installing the PEM server and PEM-HTTPD on separate hosts	101
7.5.3	Installing the PEM server on an existing Postgres servers	104
7.6	Creating an EDB repository on an isolated network	109
7.7	Configuring the PEM server on Linux	110
8	Installing Postgres Enterprise Manager agent	111
8.1	Installing Postgres Enterprise Manager agent on Linux x86 (amd64)	112
8.1.1	Installing Postgres Enterprise Manager agent on RHEL 9 or OL 9 x86_64	113
8.1.2	Installing Postgres Enterprise Manager agent on RHEL 8 or OL 8 x86_64	114
8.1.3	Installing Postgres Enterprise Manager agent on AlmaLinux 9 or Rocky	114
8.1.4	Installing Postgres Enterprise Manager agent on AlmaLinux 8 or Rocky	115
8.1.5	Installing Postgres Enterprise Manager agent on SLES 15 x86_64	116
8.1.6	Installing Postgres Enterprise Manager agent on SLES 12 x86_64	117
8.1.7	Installing Postgres Enterprise Manager agent on Ubuntu 22.04 x86_64	118
8.1.8	Installing Postgres Enterprise Manager agent on Ubuntu 20.04 x86_64	118
8.1.9	Installing Postgres Enterprise Manager agent on Debian 11 x86_64	119
8.2	Installing Postgres Enterprise Manager agent on Linux IBM Power (ppc64le)	120
8.2.1	Installing Postgres Enterprise Manager agent on RHEL 9 ppc64le	120
8.2.2	Installing Postgres Enterprise Manager agent on RHEL 8 ppc64le	121
8.2.3	Installing Postgres Enterprise Manager agent on SLES 15 ppc64le	122
8.2.4	Installing Postgres Enterprise Manager agent on SLES 12 ppc64le	123
8.3	Installing a PEM agent on Windows	124
9	Upgrade and migration	125
9.1	Upgrading a PEM installation	126
9.1.1	Upgrading a PEM native package installation on a Linux host	126
9.1.2	Upgrading a PEM installation on a Windows host	129
9.2	Upgrading the PEM backend Postgres database	131
9.3	Upgrading SQL Profiler	139
9.3.1	Upgrading a SQL profiler native package installation on a Linux host	140
9.3.2	Upgrading a SQL Profiler installation on a Windows host	141
9.4	Moving the PEM server	143
10	Uninstalling	151
10.1	Uninstalling Postgres Enterprise Manager components on Linux	151
10.2	Uninstalling Postgres Enterprise Manager components on Windows	152
11	Troubleshooting server installation	152
12	Changing the default port	154
13	Registering a Postgres server	155
14	Registering a PEM agent	163
15	Managing a PEM server	167
16	Managing a PEM agent	177
17	Managing Postgres servers	185
18	Managing certificates	187
19	Managing configuration settings	191
20	Troubleshooting agent issues	191
21	Accessing the web interface	192
22	Monitoring performance	198
22.1	Probes	200
22.2	Alerts	215
22.3	Notifications	233
22.4	Capacity Manager	239
22.5	Audit Manager	244
22.6	Log Manager	248
22.7	Charts	253
22.8	Dashboards	256
22.9	Remote monitoring	260
23	Monitoring Barman	261

24	Monitoring EDB Postgres Distributed	263
25	Monitoring Failover Manager	264
26	Monitoring Replication Server	266
27	Tuning performance	266
27.1	Tuning Wizard	266
27.2	Postgres Expert	268
27.3	Performance Diagnostic	270
28	Profiling workloads	273
28.1	SQL Profiler	274
28.1.1	Installing the SQL Profiler plugin	274
28.1.2	SQL profiling and analysis	277
28.1.3	Uninstalling SQL Profiler	279
28.2	Using the Index Advisor	280
29	Viewing system reports	280
30	PEM command line interface	282
31	Using the Query tool	285
32	Using the Schema Diff tool	291
33	Using the ERD tool	292
34	Postgres Enterprise Manager	297
34.1	PEM Getting Started	297
34.1.1	PEM Architecture	298
34.1.2	PEM Server Logon	302
34.1.3	Managing Configuration Settings	304
34.1.4	Roles for managing PEM	319
34.1.5	The Group Dialog	322
34.1.6	Automatic Discovery of Servers	322
34.1.7	Defining a server	324
34.1.8	Defining and Monitoring Postgres instances on AWS	330
34.1.9	Connect to server	332
34.1.10	Controlling a Server	333
34.1.11	Connection error	333
34.1.12	Enable Kerberos Authentication	335
34.2	Managing a PEM Agent	336
34.2.1	PEM Agent Properties	336
34.2.2	Binding an Agent to a Server	339
34.2.3	Controlling the PEM Agent	341
34.2.4	High Availability Integration	341
34.2.5	PEM Agent Privileges	342
34.2.6	PEM Agent Configuration Parameters	343
34.2.7	PEM Agent Self Registration	346
34.2.8	Register/Unregister database server using PEM Agent	347
34.3	The PEM Client	349
34.3.1	PEM Main Browser Window	349
34.3.2	Browser Toolbar	360
34.3.3	The PEM Menu Bar	362
34.3.4	PEM Client Preferences	366
34.3.5	Keyboard Shortcuts	387
34.3.6	Search objects	390
34.4	Enterprise Management Features	391
34.4.1	Dashboards	391
34.4.1.1	The Alerts Dashboard	394
34.4.1.2	The Audit Log Analysis Dashboard	396
34.4.1.3	PGD Monitoring Dashboards	398
34.4.1.4	The Database Analysis Dashboard	403
34.4.1.5	The Global Overview Dashboard	406
34.4.1.6	The I/O Analysis Dashboard	408

34.4.1.7	The Memory Analysis Dashboard	411
34.4.1.8	The Objects Activity Analysis Dashboard	412
34.4.1.9	The Operating System Analysis Dashboard	416
34.4.1.10	The Probe Log Analysis Dashboard	419
34.4.1.11	The Server Analysis Dashboard	420
34.4.1.12	The Server Log Analysis Dashboard	423
34.4.1.13	The Session Activity Analysis Dashboard	425
34.4.1.14	The Session Waits Analysis Dashboard	426
34.4.1.15	The Storage Analysis Dashboard	427
34.4.1.16	The System Wait Analysis Dashboard	429
34.4.1.17	The Streaming Replication Analysis Dashboard	430
34.4.2	Server Configuration	433
34.4.2.1	Configuration Options	434
34.4.3	Log Manager	439
34.4.4	Audit Manager	448
34.4.5	Postgres Log Analysis Expert	453
34.4.6	Tuning Wizard	460
34.4.7	Postgres Expert	467
34.4.7.1	Schema Expert Recommendations	471
34.4.7.2	Security Expert Recommendations	473
34.4.7.3	Configuration Expert Recommendations	474
34.4.8	Capacity Manager	479
34.4.8.1	Tab 1 (Metrics)	484
34.4.8.2	Tab 2 (Options)	485
34.4.9	Alerting	487
34.4.9.1	Creating and Managing Alerts	490
34.4.9.2	Copy Alerts	500
34.4.9.3	Alert Templates	501
34.4.9.4	Custom Alert Templates	509
34.4.9.5	Email Groups	515
34.4.9.6	Webhooks	517
34.4.9.7	SNMP MIB Generation	521
34.4.9.8	SNMP Trap Details	522
34.4.9.9	Using PEM with Nagios	523
34.4.10	Using the Manage Charts Tab	527
34.4.10.1	Creating a New Chart	528
34.4.10.2	Importing a Capacity Manager Template	531
34.4.11	The PEM Manage Dashboards Tab	538
34.4.11.1	Creating a Custom Dashboard	539
34.4.11.2	Creating an Ops Dashboard	545
34.4.12	The Manage Probes Tab	546
34.4.12.1	Custom Probes	547
34.4.12.2	Copy Probe Configuration	553
34.4.12.3	Probe Configuration	554
34.4.12.3.1	Probes	556
34.4.13	Schedule Alert Blackout	563
34.4.14	Scheduled System Jobs	567
34.4.15	Scheduled Task Tab	567
34.4.16	Creating a PEM Scheduled Job	569
34.4.17	Sending email notifications for a job	578
34.4.18	Task Viewer	581
34.4.18.1	Log Viewer	582
34.4.19	Monitoring a Failover Manager Cluster	582
34.4.20	Monitoring a PGD Group	584
34.4.21	Performance Diagnostic	585
34.4.22	Reports	590

34.5	Management Basics	594
34.5.1	Grant Wizard	595
34.5.2	Add named restore point Dialog	597
34.5.3	Import/Export data Dialog	598
34.5.4	Maintain a database object	601
34.5.4.1	Maintenance Dialog	602
34.5.5	Storage Manager	603
34.5.6	Backup Dialog	605
34.5.7	Backup Globals Dialog	610
34.5.8	Backup Server Dialog	612
34.5.9	Restore Dialog	616
34.5.10	Managing Cluster Level Objects	621
34.5.10.1	Database Dialog	621
34.5.10.2	Move Objects Dialog	627
34.5.10.3	Resource Group Dialog	629
34.5.10.4	Login/Group Role Dialog	631
34.5.10.5	Tablespace Dialog	641
34.5.10.6	Role Reassign/Drop Own Dialog	646
34.5.11	Managing Database Objects	650
34.5.11.1	Cast Dialog	650
34.5.11.2	Collation Dialog	653
34.5.11.3	Domain Dialog	656
34.5.11.4	Domain Constraints Dialog	661
34.5.11.5	Event Trigger Dialog	664
34.5.11.6	Extension Dialog	668
34.5.11.7	Foreign Data Wrapper Dialog	671
34.5.11.8	Foreign Server Dialog	676
34.5.11.9	Foreign Table Dialog	681
34.5.11.10	FTS Configuration dialog	688
34.5.11.11	FTS Dictionary Dialog	692
34.5.11.12	FTS Parser Dialog	696
34.5.11.13	FTS Template Dialog	699
34.5.11.14	Function Dialog	702
34.5.11.15	Language Dialog	708
34.5.11.16	Materialized View Dialog	711
34.5.11.17	Package Dialog	717
34.5.11.18	Procedure Dialog	721
34.5.11.19	Publication Dialog	727
34.5.11.20	Schema Dialog	729
34.5.11.21	Sequence Dialog	733
34.5.11.22	Subscription Dialog	737
34.5.11.23	Synonym Dialog	741
34.5.11.24	Trigger function Dialog	743
34.5.11.25	Type Dialog	750
34.5.11.26	User Mapping Dialog	758
34.5.11.27	View Dialog	761
34.5.12	Creating or Modifying a Table	766
34.5.12.1	Check Dialog	766
34.5.12.2	Column Dialog	769
34.5.12.3	Compound Trigger Dialog	777
34.5.12.4	Exclusion constraint Dialog	782
34.5.12.5	Foreign key Dialog	786
34.5.12.6	Index Dialog	791
34.5.12.7	Primary key Dialog	795
34.5.12.8	RLS Policy Dialog	797
34.5.12.9	Rule Dialog	800

34.5.12.10	Table Dialog	805
34.5.12.11	Trigger Dialog	822
34.5.12.12	Unique Constraint Dialog	828
34.6	Monitoring Barman	831
34.6.1	Prerequisites for managing Barman	832
34.6.2	Configuring a Barman Server	832
34.6.3	Viewing the Barman Server details on a PEM dashboard	835
34.7	SQL Profiler	838
34.7.1	Installing SQL Profiler	838
34.7.2	Configuring SQL Profiler	845
34.7.3	Using SQL Profiler	845
34.7.4	Using Index Advisor	851
34.7.5	The SQL Profiler Tab	852
34.8	Developer Tools	854
34.8.1	PL Debugger	854
34.8.2	Query tool	859
34.8.3	Interpreting Graphical Query Plans	875
34.8.4	Reviewing and Editing Data	878
34.8.4.1	View/Edit Data Filter	884
34.8.5	Schema Diff	884
34.8.6	ERD Tool	888
34.9	Configuring pgBouncer for use with PEM Agents	895
34.9.1	Connecting PEM to pgBouncer	896
34.9.2	Preparing the PEM Server for pgBouncer Connections	897
34.9.3	Configuring pgBouncer	898
34.9.4	Configuring the PEM Agent to use pgBouncer	900
34.10	pgAgent	901
34.10.1	Using pgAgent	902
34.10.2	Installing pgAgent	902
34.10.3	Creating a pgAgent Job	904
34.10.4	pgAgent Steps	911
34.10.5	pgAgent Schedules	912
34.11	Appendices	916
34.11.1	Licence	916
34.11.2	The MIT Kerberos Licence	916
34.11.3	The OpenSSL Licence	917
34.11.4	The SNMP++ Licence	918
34.11.5	The jquery table sort Licence	918
34.12	PEM Worker Usage Guide	918

# 1 Postgres Enterprise Manager

Welcome to Postgres Enterprise Manager (PEM). PEM consists of components that provide the management and analytical functionality for your EDB Postgres Advanced Server or PostgreSQL database. PEM is based on the Open Source pgAdmin 4 project.

PEM is a comprehensive database design and management system. PEM is designed to meet the needs of both novice and experienced Postgres users alike, providing a powerful graphical interface that simplifies the creation, maintenance, use of database objects and monitoring multiple postgres servers through a single graphical interface.

## Postgres compatibility

Supported versions of Postgres for PEM 8.x:

	Monitored Instance	Backend Instance
EDB Postgres Advanced Server (EPAS)	12, 13, 14	12, 13, 14
PostgreSQL (PG)	12, 13, 14	12, 13, 14
EDB Postgres Extended Server (PGE)	12, 13, 14	13, 14 <sup>[^1]</sup>

[^1]: sslutils isn't available for RHEL 7 on IBM Power, so this distribution can't use PGE as a backend.

# 2 Release notes

The Postgres Enterprise Manager (PEM) documentation describes the latest version of PEM 8 including minor releases and patches. The release notes in this section provide information on what was new in each release. For new functionality introduced after 8.0.0 in a minor or patch release, there are also indicators within the content about what release introduced the feature.

Version	Release Date	Upstream Merges	Accessibility Conformance
<a href="#">8.7.1</a>	04 Mar 2024	NA	<a href="#">Conformance Report</a>
<a href="#">8.7.0</a>	29 Nov 2023	NA	<a href="#">Conformance Report</a>
<a href="#">8.6.4</a>	12 Oct 2023	NA	<a href="#">Conformance Report</a>
<a href="#">8.6.3</a>	14 Jul 2023	NA	<a href="#">Conformance Report</a>
<a href="#">8.6.1</a>	31 Mar 2023	NA	<a href="#">Conformance Report</a>
<a href="#">8.6.0</a>	29 Sep 2022	NA	<a href="#">Conformance Report</a>
<a href="#">8.5.0</a>	23 Jun 2022	NA	<a href="#">Conformance Report</a>
<a href="#">8.4.0</a>	01 Apr 2022	NA	<a href="#">Conformance Report</a>
<a href="#">8.3.0</a>	24 Nov 2021	pgAdmin <a href="#">5.7</a>	<a href="#">Conformance Report</a>
<a href="#">8.2.0</a>	09 Sep 2021	pgAdmin <a href="#">5.4</a> , <a href="#">5.5</a> , and <a href="#">5.6</a>	<a href="#">Conformance Report</a>
<a href="#">8.1.1</a>	22 Jul 2021	NA	<a href="#">Conformance Report</a>
<a href="#">8.1.0</a>	16 Jun 2021	pgAdmin <a href="#">5.0</a> , <a href="#">5.1</a> , <a href="#">5.2</a> , and <a href="#">5.3</a>	<a href="#">Conformance Report</a>
<a href="#">8.0.1</a>	3 Mar 2021	pgAdmin <a href="#">4.29</a> , <a href="#">4.30</a> , and <a href="#">5.0</a>	NA
<a href="#">8.0.0</a>	9 Dec 2020	pgAdmin <a href="#">4.27</a> , <a href="#">4.28</a> , and <a href="#">4.29</a>	<a href="#">Conformance Report</a>

Often only select issues are included in the upstream merges. The specific issues included in the merges are listed in the release note topics.

## 2.1 Postgres Enterprise Manager 8.7.1 release notes

Released: 04 Mar 2024

New features, enhancements, bug fixes, and other changes in PEM 8.7.1 include:

Type	Description
Security fix	This is a security fix for <a href="#">CVE-2023-50782</a> . Upgraded bundled Python cryptography package to 42.x.
Security fix	This is a security fix for <a href="#">CVE-2023-31122</a> and <a href="#">CVE-2023-43622</a> . Upgraded bundled HTTPD installer to 2.4.58 for PEM Windows installer to fix Apache HTTPD server issues.

## 2.2 Postgres Enterprise Manager 8.7.0 release notes

Released: 29 Nov 2023

New features, enhancements, bug fixes, and other changes in PEM 8.7.0 include:

Type	Description
Enhancement	Upgraded the PEM agent windows installer to x64.
Security fix	This is a security fix for <a href="#">CVE-2023-38545</a> and <a href="#">CVE-2023-38546</a> . This patch fixes the libcurl-pem issues.

## 2.3 Postgres Enterprise Manager 8.6.4 release notes

Released: 12 Oct 2023

New features, enhancements, bug fixes, and other changes in PEM 8.6.4 include:

Type	Description
Security fix	This is a security fix for <a href="#">CVE-2023-5002</a> . This security fix includes updates for the pgAdmin vulnerability issues. This patch affects the PEM server only, no need to update PEM agents.
Security fix	This is a security fix for <a href="#">CVE-2023-4863</a> . This security fix includes updates for the libwebp issues. This patch affects the PEM server only, no need to update PEM agents.

## 2.4 Version 8.6.3

New features, enhancements, bug fixes, and other changes in PEM 8.6.3 include:

Type	Description
Security fix	Added support for Flask 2.x. This is a security fix for CVE-2023-30861 and is recommended for all users. This patch affects the PEM server only, no need to update PEM agents.

## 2.5 Version 8.6.1

New features, enhancements, bug fixes, and other changes in PEM 8.6.1 include:

Type	Description	ID
Bug fix	Fixed an issue whereby an error would occur when registering a PEM agent on Ubuntu 22.04	PEM-4517

## 2.6 Version 8.6.0

New features, enhancements, bug fixes, and other changes in PEM 8.6.0 include:

Type	Description	ID
Enhancement	Added the ability for PEM agent to allow connections to use SSL passwords for greater interoperability with multi-domain SSL certificates and compliance satisfaction [Support Tickets #73628, #82574, #82144, #82175]	PE M-4276
Enhancement	Improved the team support for the database servers, agents, and tools. When no team is specified for the particular object, you can control the visibility of the database servers, agents, and tools by configuring the <code>show_objects_with_no_team</code> parameter. [Support Ticket #922716]	PE M-2843
Enhancement	Added the ability to differentiate the Postgres Extended servers in the core usage report.	PE M-4212
Enhancement	Added support for monitoring the PostgreSQL 15 (beta).	PE M-4544
Bugfix	Fixed an issue parsing the BARMAN <code>diagnose</code> API.	PE M-4517
Bugfix	Fixed an issue that was resulting in an error. This error occurred during a purging operation for the probes data of the <code>extension</code> type.	PE M-4576

## 2.7 Version 8.5.0

New features, enhancements, bug fixes, and other changes in PEM 8.5.0 include:



Type	Description	ID
New Feature	Added a command-line utility to import/export the custom probes and alert templates into/from the PEM Servers. See <a href="#">PEM command line interface</a> for more information.	PEM-4206
Bug Fix	Fixed the "duplicate key value violates unique constraint 'autodiscoverserverpkey'" error. [Support Ticket #RT75870]	PEM-4385
Bug Fix	Fixed an issue in the configure script (*nix), which did not allow configuration of the PEM server without the 'superuser' privilege.	PEM-4468

## 2.8 Version 8.4.0

New features, enhancements, bug fixes, and other changes in PEM 8.4.0 include:

Type	Description	ID
New Feature	Built-in support for monitoring Barman backups. See <a href="#">Monitoring Barman</a> for more information.	PEM-4435
Security Fix	Hardened against unrestricted file uploads as reported in the medium severity CVE-2022-0959.	PEM-4442
Enhancement	Monitoring of transaction ID (TXID) wraparound for exhaustion and to prevent failure.	PEM-3990
Enhancement	Removed unnecessary monitoring of virtual file systems. [Support Ticket #573096]	PEM-806
Enhancement	Sorting based on status for agent and server tables in the dashboard.	PEM-4152
Enhancement	Option to disable the Query tool for users in order to restrict viewing data. [Support Ticket #74976]	PEM-4315
Enhancement	Support for Postgres extension-based probes for multi-version flexibility. Updating to newer versions are no longer required.	PEM-4391
Enhancement	Improved the Linux installation instructions with added details and steps.	PEM-4381
Bug Fix	A new installation grants the <code>pem_admin</code> role to the superuser. [Support Ticket #79577]	PEM-4433
Bug Fix	For display tables, numeric fields sorted by numeric order, not alphabetical order. [Support Ticket #1111704]	PEM-3827
Bug Fix	Limits the decimal precision displayed for monitoring percentages.	PEM-4144
Bug Fix	Duplicate key value violates unique constraint blocked_session_info_pkey. [Support Ticket #RT75870]	PEM-4333
Bug Fix	Probe error for Postgres Extended 14.	PEM-4356
Bug Fix	PEM agent not gathering data after upgrade. [Support Ticket #78679]	PEM-4430
Bug Fix	Added an option in preferences to change the line ending of the email body content from LF(Line Feed) to CRLF (Carriage Return Line Feed). This fixes missing alert body content in email notifications. [Support Ticket #833910]	PEM-1832
Bug Fix	Fixed pg_isready time out issue in configure script [Support Ticket #RT75651]	PEME SC-386

## 2.9 Version 8.3.0

New features, enhancements, bug fixes, and other changes in PEM 8.3.0 include:

Type	Description	ID
New Feature	New roles, pem_manage_dashboard and pem_manage_chart for creating and managing charts and dashboards are added.	PEM-728
Enhancement	Supports proxy in the PEM agent webhook configuration [Support Ticket #72432].	PEM-4185
Enhancement	Support Export/Import Custom charts and Custom Dashboards using PEM GUI and RESTapi.	PEM-4300, PEM-4325, PEM-3443, PEM-4215
Enhancement	Mask the username from the login related errors when displaying on GUI and return generic error message. [Support Ticket #1341245]	PEM-4150
Enhancement	Support for PostgreSQL, EDB Postgres Advanced Server and EDB Postgres Extended Server version 14 is added.	PEM-4091
Enhancement	Support for monitoring status and activities for pgd.stat_subscription and pgd.stat_relation on PGD (Bi-Directional Replication) dashboards and probes is added.	PEM-4179
Bug Fix	Support for the truncate table with restart identity.	<a href="#">#2538</a>
Bug Fix	Make code folding case insensitive in the code mirror.	<a href="#">#4264</a>
Bug Fix	The database and server information displays on the maintenance process watcher dialog.	<a href="#">#4629</a>
Bug Fix	Allow the referenced table to be the same as the local table in one to many relationship for ERD Tool.	<a href="#">#6495</a>
Bug Fix	Make closing tabs to be smarter by focusing on the appropriate tab when the user closed a tab.	<a href="#">#6625</a>
Bug Fix	Set PSQLRC and PSQL_HISTORY environment variables to any specific path to have user separate psql history files.	<a href="#">#6691</a>
Bug Fix	Table chart rendering for numeric metrics is enhanced. [Support Ticket #880817]	PEM-2251
Bug Fix	A notification banner displays on the monitoring panel when user tries to access excluded database dashboard.	PEM-4218
Bug Fix	User can use the same internal SMTP TLS settings for 2FA & Email alerts both.	PEM-4262
Bug Fix	The maxlength of url & payload to allow long values in webhook module is increased. [Support Ticket #75367]	PEM-4326
Bug Fix	The SQL query to revoke privileges on objects like tables are correctly displayed in SQL tab.	<a href="#">#4567</a>
Bug Fix	The user can paste the updated table header in safari 12 and 13 browsers.	<a href="#">#4815</a>
Bug Fix	Ensure that trigger function SQL should have 'create or replace function' instead of 'create function' only.	<a href="#">#5849</a>
Bug Fix	The images can be exported for the relations where the lines are over the nodes.	<a href="#">#6531</a>
Bug Fix	The columns with sequences do not get altered unnecessarily with a schema diff tool.	<a href="#">#6564</a>
Bug Fix	The Decimal number is appended for character varying fields while downloading the data in CSV format.	<a href="#">#6572</a>
Bug Fix	The data output panel display is partially fixed.	<a href="#">#6536</a>
Bug Fix	The whitespace in function bodies are applied while generating the script using schema diff tool.	<a href="#">#6620</a>
Bug Fix	Enables PEM to retrieve user permissions in case of nested roles which helps to terminate the session for AWS RDS.	<a href="#">#6641</a>
Bug Fix	The errors related to HTML tags shown in the error message for JSON editor are fixed.	<a href="#">#6668</a>
Bug Fix	Fixed UnboundLocalError where local variable 'user_id' referenced before assignment.	<a href="#">#6671</a>
Bug Fix	Renamed 'Auto rollback?' to 'Auto rollback on error?'.	<a href="#">#6682</a>
Bug Fix	The JSON editor issue of hiding the first record.	<a href="#">#6684</a>
Bug Fix	Ensure that deleting a database should not automatically connect to the next database.	<a href="#">#6685</a>
Bug Fix	Multiple query tool tabs does not get closed for the single close event.	<a href="#">#6710</a>

## 2.10 Version 8.2.0

New features, enhancements, bug fixes, and other changes in PEM 8.2.0 include:

Type	Description	ID
New Feature	Two-Factor Authentication for PEM Server to decrease risk of unauthorized access. See <a href="#">Configuring the PEM server to use two-factor authentication</a> .	PEM-3680, PEM-4090
New Feature	View, edit, and format JSON data rendered in an effective way in the Query tool.	<a href="#">#5198</a>
Enhancement	Ability to exclude a particular database from PEM monitoring [Support Ticket #1196669]. See <a href="#">Exclude a database from PEM Monitoring</a> .	PEM-3824
Enhancement	Support Export/Import Custom Probes and Custom Alert Templates using PEM GUI and RESTapi.	PEM-4160, PEM-4183
Enhancement	Highlight the long running queries on the dashboards.	<a href="#">#1975</a>
Enhancement	Do not block the query editor window while running a query.	<a href="#">#3920</a>
Enhancement	Copy SQL from main window to query tool.	<a href="#">#4904</a>
Enhancement	Set the binary path for the different database server versions.	<a href="#">#5370</a>
Enhancement	Provide maximum width of the column when 'Resize by data' option in the preferences is set to True.	<a href="#">#6559</a>
Bug Fix	New ids are updated to probe schedule table after re-registering agent/server. [Support Ticket #1324590]	PEM-4178
Bug Fix	Query tool can snap the Data Output panel back after it is detached.	<a href="#">#4189</a>
Bug Fix	Replace the keyboard shortcut issue in the query tool on the standard keyboard layout.	<a href="#">#6388</a>
Bug Fix	Detaching the query editor panel gives a blank white panel.	<a href="#">#6398</a>
Bug Fix	The Query tool does not change the connection string as per the <a href="#">Query tool tab title</a> .	<a href="#">#6404</a>
Bug Fix	Remove leading whitespace and replace it with '['...' in the Query tool data grid, so cells don't look empty.	<a href="#">#6427</a>
Bug Fix	Users can search in <a href="#">all types</a> or <a href="#">subscriptions</a> in search objects without having access to subscriptions.	<a href="#">#6448</a>
Bug Fix	Users can add members while creating a Login/Role group.	<a href="#">#6466</a>
Bug Fix	The properties panel of the Role displays the calendar.	<a href="#">#6469</a>
Bug Fix	Execute/Refresh button should not be disabled when we run the empty query.	<a href="#">#6489</a>
Bug Fix	New Connection Drop Down has lost default maintenance database, auto-select, and tab-through functionality.	<a href="#">#6505</a>
Bug Fix	The Decimal number is appended for character varying fields while downloading the data in CSV format.	<a href="#">#6520</a>
Bug Fix	The Directory selection issue with the folder dialog.	<a href="#">#6536</a>
Bug Fix	The setting 'Open in new browser tab' is visible and not based on the value of 'ENABLE_PSQL'.	<a href="#">#6541</a>
Bug Fix	Copy/paste issues for PSQL tool terminal.	<a href="#">#6547</a>
Bug Fix	Disable email deliverability check that was introduced in flask-security-too by default to maintain backward compatibility.	<a href="#">#6550</a>
Bug Fix	Incorrect column name listed in the properties of Index.	<a href="#">#6557</a>
Bug Fix	Paste is not working through Right-Click option on PSQL.	<a href="#">#6574</a>
Bug Fix	TypeError 'NoneType' object is not sub scriptable.	<a href="#">#6580</a>
Bug Fix	Incorrect tablespace options in the drop-down for move objects dialog.	<a href="#">#6586</a>
Bug Fix	Titles in query tabs are different.	<a href="#">#6618</a>
Bug Fix	Incorrect binary path issue when the user deletes the binary path from the preferences.	<a href="#">#6619</a>
Bug Fix	Ensure that all the required options should be loaded when the Range data type is selected while creating a custom data type.	<a href="#">#6643</a>
Bug Fix	Dashboard server activity issue when active_since parameter is None.	<a href="#">#6650</a>

## 2.11 Version 8.1.1

The bug fix in PEM 8.1.1 is:

Type	Description	ID
Bug Fix	REST API tokens are being generated but clients are unable to access the endpoints when using the valid tokens in a service. [Support Ticket: RT71632]	PEM-3893

## 2.12 Version 8.1.0

New features, enhancements, bug fixes, and other changes in PEM 8.1.0 include:

Type	Description	ID
New Feature	PGD (Bi-Directional Replication) dashboards and probes to monitor status and activities for admin, nodes, and groups. For example, if replication starts falling behind, the administrator is alerted and can take proactive action. See <a href="#">Monitoring EDB Postgres Distributed</a> .	PEM-3893
New Feature	Kerberos security and authentication for PEM Web Client through GSSAPI or Active Directory Domain Services. Prevent password interception and prove identity required by highly regulated industries. [Support Ticket #1044678] See <a href="#">Configuring the PEM server to use Kerberos authentication</a> .	PEM-3559
New Feature	Entity-Relationship Diagram (ERD) design tool that graphically represents database tables, columns, and interrelationships. Conceptualize database design before building or reverse engineering. See <a href="#">The ERD Tool</a> .	<a href="#">#6397</a>
New Feature	Logical replication support to manage publications and subscriptions for core logical replication in Postgres. Send incremental changes across different database versions or platforms.	<a href="#">#5912</a>
New Feature	Quick Search option to locate features and relevant help articles with a guided user experience. Type three characters to start displaying matching possibilities.	<a href="#">#6148</a>
Enhancement	EDB Postgres Extended Server (used by PGD Enterprise) certified for monitoring and administration.	PEM-4015
Enhancement	Configuration parameters for Audit Manager, Tuning Wizard, and Log Manager no longer require a restart of the database. [Support Ticket #649078]	PEM-795, PEM-1465, PEM-3829
Enhancement	Manage the privileges for a package node with Grant Wizard.	<a href="#">#1591</a>
Enhancement	Maximize/restore the window for the properties dialog.	<a href="#">#4064</a>
Enhancement	While examining the group role, display the login roles.	<a href="#">#5404</a>
Enhancement	Set automatic width of columns by content size in the data output window.	<a href="#">#5954</a>
Enhancement	Support for cache busting webpack chunk files.	<a href="#">#5477</a>
Enhancement	SSL support for creating a subscription.	<a href="#">#6201</a>
Enhancement	Support for the creation of Nested Table and Varying Array Type for Advanced Server.	<a href="#">#6407</a>
Enhancement	Comment column in the properties panel for View and Materialized View collection node.	<a href="#">#6416</a>
Bug Fix	Process the alert if there is an error while processing alert detailed information. [Support Ticket #1201865]	PEM-1032
Bug Fix	Remove the obsolete data of a deleted database from pemhistory on the next purge. Support Ticket #1055978]	PEM-3810
Bug Fix	Flapping does not end when at least a one-time alert triggered during the flapping period. [Support Ticket #1165155]	PEM-3966
Bug Fix	Getting 404 for REST API of agent probe history. [Support Ticket #1219795]	PEM-3974
Bug Fix	The "A user expires in N days" alert template code failing on Postgres. [Support Ticket #1235289]	PEM-3988
Bug Fix	Check for sslutils for version 1.3 in the configure script. [Support Ticket #1284690]	PEM-4042
Bug Fix	Correct OID for binding variable "downObjects" for AgentsDown and ServersDown alerts. [Support Ticket #1300113]	PEM-4073
Bug Fix	Read the log configuration data from the system probe tables for the selected servers in the Log Manager. [Support Ticket #1262817]	PEM-4029
Bug Fix	Skip the PEM Server registration when installing using only a web server option to avoid duplicate entries in the server listing. [Support Ticket #1210935]	PEM-3977
Bug Fix	Alignment issues under preferences for the German language.	<a href="#">#4014</a>
Bug Fix	Issue with renaming the database by another user.	<a href="#">#4203</a>
Bug Fix	Drag and drop object is not correct in CodeMirror for properties dialog.	<a href="#">#4436</a>
Bug Fix	Adding/updating records fails if the table name contains a percent sign.	<a href="#">#4438</a>

Type	Description	ID
Bug Fix	Ensure that autovacuum and analyze scale factors should be editable with more than two decimals.	<a href="#">#4784</a>
Bug Fix	% displayed twice in Explain Analyze for query and table.	<a href="#">#4847</a>
Bug Fix	Rename text 'table' to 'relation' in the Statistics tab for Explain Analyze.	<a href="#">#4849</a>
Bug Fix	Properties tab for Collection Nodes is unresponsive after switching the tabs.	<a href="#">#4959</a>
Bug Fix	Save button is enabled for functions/procedures by default when opening the Properties dialog.	<a href="#">#5073</a>
Bug Fix	Hanging symlinks in a directory causes Select file dialog to break.	<a href="#">#5119</a>
Bug Fix	Ensure that the Query tool tab should be closed after server disconnection when auto-commit/auto-rollback is set to false.	<a href="#">#5519</a>
Bug Fix	Data is displayed in the wrong order when executing the query repeatedly.	<a href="#">#5555</a>
Bug Fix	Ensure that cell content is auto-selected when editing the cell data.	<a href="#">#5810</a>
Bug Fix	Ensure that SQL formatter should not add extra tabs and format the SQL correctly.	<a href="#">#5869</a>
Bug Fix	Shortcut keys are not working with manage macro.	<a href="#">#5908</a>
Bug Fix	Encoding issue when database encoding set to SQL_ASCII and name of the column is in ASCII character.	<a href="#">#6018</a>
Bug Fix	Ensure that the user should not change the connection when a long query is running.	<a href="#">#6082</a>
Bug Fix	Flickering issue with the input box on check constraints.	<a href="#">#6106</a>
Bug Fix	User is unable to update column-level privileges from the security tab.	<a href="#">#6117</a>
Bug Fix	Ensure that the user should be able to kill the session from Dashboard if the user has a 'pg_signal_backend' role.	<a href="#">#6159</a>
Bug Fix	Cursor shifts its focus to the wrong window for all the Query tool related model dialogs.	<a href="#">#6161</a>
Bug Fix	Correct the syntax for 'CREATE TRIGGER', use 'EXECUTE FUNCTION' instead of 'EXECUTE PROCEDURE' from Postgres v11 onwards.	<a href="#">#6220</a>
Bug Fix	User is not able to create the subscription.	<a href="#">#6230</a>
Bug Fix	Ensure that SQL formatter should not use tab size if 'Use spaces?' set to false.	<a href="#">#6233</a>
Bug Fix	Ensure that the view/edit data panel should not be opened for unsupported nodes using the keyboard shortcut.	<a href="#">#6206</a>
Bug Fix	User is unable to create a subscription if the host/IP address for connection is 127.0.0.1.	<a href="#">#6253</a>
Bug Fix	Ensure that proper error messages should be shown on the properties and statistics tab in case of insufficient privileges for a subscription.	<a href="#">#6259</a>
Bug Fix	'Create Slot' option is disabled in case of the same IP/host provided but the port is different.	<a href="#">#6260</a>
Bug Fix	User is not able to change the connection in the Query tool when any SQL file is opened.	<a href="#">#6272</a>
Bug Fix	Ensure that the strings in the LDAP auth module are translatable.	<a href="#">#6274</a>
Bug Fix	Schema diff shows the wrong SQL when comparing triggers with different when clauses.	<a href="#">#6281</a>
Bug Fix	Ensure that the template database should be visible while creating the database.	<a href="#">#6286</a>
Bug Fix	String index out of range error where the dependent tab is in focus and selecting any publication or table.	<a href="#">#6292</a>
Bug Fix	Procedure creation failed when providing the Volatility option.	<a href="#">#6293</a>
Bug Fix	Dependent tab threw an error when selecting any Login/Group role.	<a href="#">#6294</a>
Bug Fix	While selecting the row which was deleted just before the selection operation.	<a href="#">#6306</a>
Bug Fix	Incorrect values were visible in the Dependents tab for publication.	<a href="#">#6307</a>
Bug Fix	Copy/paste rows in view data paste the wrong value for Boolean type.	<a href="#">#6312</a>
Bug Fix	Ensure that the primary key should be visible properly in the Table dialog.	<a href="#">#6316</a>
Bug Fix	Ensure that toggle buttons are accessible by most screen readers.	<a href="#">#6317</a>
Bug Fix	Ensure that the grantor name should be visible properly for the Security tab in the Table dialog.	<a href="#">#6323</a>
Bug Fix	Ensure that the file format for the storage manager should be 'All files' and for other dialogs, it should remember the last selected format.	<a href="#">#6325</a>
Bug Fix	Ensure that while comparing domains check function dependencies should be considered in schema diff.	<a href="#">#6327</a>
Bug Fix	Wrong SQL is showing for the child partition tables.	<a href="#">#6329</a>
Bug Fix	Sizing issue of help dialog for Query tool when open in the new browser tab.	<a href="#">#6333</a>
Bug Fix	CSV download quotes the numeric columns.	<a href="#">#6341</a>

Type	Description	ID
Bug Fix	Cannot unpack non-iterable response object error when selecting any partition.	<a href="#">#6344</a>
Bug Fix	Ensure that we should not allow opening external files that are dragged into it.	<a href="#">#6355</a>
Bug Fix	Users are unable to see data of the partitions using the View/Edit data option.	<a href="#">#6375</a>
Bug Fix	A connection warning should be displayed when the user clicks on Explain or Explain Analyze and the database server is disconnected from the browser tree.	<a href="#">#6376</a>
Bug Fix	Schema diff does not create DROP DEFAULT statements for columns.	<a href="#">#6377</a>
Bug Fix	Foreign data wrapper properties are not visible if the host option contains two host addresses.	<a href="#">#6379</a>
Bug Fix	Ensure that Backup and Restore should work on shared servers.	<a href="#">#6385</a>
Bug Fix	The filter 'Include/Exclude By Selection' is not working for null values.	<a href="#">#6392</a>
Bug Fix	Ensure that the user should not be able to add duplicate panels.	<a href="#">#6399</a>
Bug Fix	Current debug line is not visible in the 'Dark' theme.	<a href="#">#6409</a>
Bug Fix	Duplicate columns are visible in the browser tree, which is owned by two sequences.	<a href="#">#6413</a>
Bug Fix	Help message was not displaying correctly on the Login/Group role.	<a href="#">#6414</a>
Bug Fix	Query editor is not being closed if the user clicks on the 'Don't Save' button.	<a href="#">#6417</a>
Bug Fix	File dialog shows incorrect files for the selected file types.	<a href="#">#6443</a>
Bug Fix	Duplicate SQL issue for tables with more than one partition.	<a href="#">#6478</a>

## 2.13 Version 8.0.1

Enhancements, bug fixes, and other changes in PEM 8.0.1 include:

Type	Description	ID
Enhancement	Allowing the dash (-) character in the superuser name that is provided during configuration. [Support Ticket #1055435]	PEM-3577
Enhancement	Gracefully closing the operating system resources during batch probe and command execution to avoid errors. [Support Ticket #1048713]	PEM-3881
Enhancement	Improved support for Unicode string handling in pemAgent. [Support Ticket #1153153]	PEM-3901
Enhancement	Option to hide or unhide the Dashboard, Statistics, Dependents, and Dependencies tabs.	<a href="#">#5091</a>
Enhancement	Improve the explain plan details by showing popup instead of tooltip on clicking of the specified node.	<a href="#">#5488</a>
Enhancement	Added appropriate help message and a placeholder for letting users know about the account password expiry for Login/Group Role.	<a href="#">#5973</a>
Enhancement	Added informative message when there is no difference found for schema diff.	<a href="#">#6122</a>
Enhancement	Added 'Count Rows' option to the partition sub tables.	<a href="#">#5282</a>
Bug Fix	Dashboard refresh not displaying BART. [Support Ticket #1046037] backups.	PEM-3549
Bug Fix	Bundle pgaevent.dll in agent Windows installer to suppress event message log error. [Support Ticket #1106021]	PEM-3845
Bug Fix	BART restore failing when the agent is not bound with the BART server. [Support Ticket #1129454]	PEM-3903
Bug Fix	Pressing the back button shows another instance of the main page inside of the Query tool tab.	<a href="#">#4892</a>
Bug Fix	The focus is not properly set on the filter text editor after closing the error dialog.	<a href="#">#5809</a>
Bug Fix	Ensure that username should be visible in the 'Connect to Server' popup when service and user name are both specified.	<a href="#">#5871</a>
Bug Fix	Ensure that the 'template1' database should not be visible after pg_upgrade.	<a href="#">#5875</a>

Type	Description	ID
Bug Fix	False error is shown while adding a new foreign key from the table dialog when a foreign key already exists and Auto FK Index set to true.	<a href="#">#5886</a>
Bug Fix	The Save button is enabled by default in Macro.	<a href="#">#5905</a>
Bug Fix	Remove the extra line after the Manage Macros menu while clearing all macros.	<a href="#">#5906</a>
Bug Fix	Ensure that 'Clear All Rows' should not work if there is no existing macro available and the user does not specify any value.	<a href="#">#5907</a>
Bug Fix	The server is disconnected with an error message displayed if the user creates Macro with invalid SQL.	<a href="#">#5929</a>
Bug Fix	Ensure that the macro query result should be downloaded properly.	<a href="#">#5965</a>
Bug Fix	Ensure that dirty indicator (*) should not be visible when renaming the tabs.	<a href="#">#5991</a>
Bug Fix	The escape character is shown when the server/database name has some special characters.	<a href="#">#5992</a>
Bug Fix	Update Flask-BabelEx to the latest.	<a href="#">#5997</a>
Bug Fix	Schema diff doesn't show the result of compare if source schema has tables with RLS.	<a href="#">#5998</a>
Bug Fix	An illegal argument is showing for trigger SQL when a trigger is created for View.	<a href="#">#6003</a>
Bug Fix	Autocomplete issue where it is not showing any suggestions if the schema name contains escape characters.	<a href="#">#6045</a>
Bug Fix	The state of the Save File icon does not match the dirty editor indicator.	<a href="#">#6046</a>
Bug Fix	The dirty indicator stays active even if all changes were undone.	<a href="#">#6047</a>
Bug Fix	Ensure that the rename panel should be disabled when the SQL file opened in the query tool.	<a href="#">#6058</a>
Bug Fix	Extra parentheses issue around joins for Views.	<a href="#">#6061</a>
Bug Fix	Accessibility issues in schema diff module.	<a href="#">#6065</a>
Bug Fix	Issue on refreshing files in Query tool.	<a href="#">#6069</a>
Bug Fix	Accessibility issues in various dialogs.	<a href="#">#6077</a>
Bug Fix	TypeError exception in schema diff when selected any identical object.	<a href="#">#6084</a>
Bug Fix	The dependencies tab showing multiple owners for the objects having shared dependencies.	<a href="#">#6087</a>
Bug Fix	Issue of deleting records when the user tries to delete multiple records.	<a href="#">#6098</a>
Bug Fix	Ensure that the user should be able to specify an older date for the account expiration of the role/user.	<a href="#">#6120</a>
Bug Fix	The database list in the new connection window is not visible.	<a href="#">#6121</a>
Bug Fix	Sequences are not created.	<a href="#">#6128</a>
Bug Fix	Ensure that verbose logs should be visible for Utility (Backup, Maintenance) jobs.	<a href="#">#6140</a>
Bug Fix	Ensure that the current value of the sequence should be ignored while comparing using schema diff.	<a href="#">#6144</a>
Bug Fix	Strikethrough is not visible for rows selected for deletion after scrolling.	<a href="#">#6157</a>
Bug Fix	User was unable to change the background color for a server.	<a href="#">#6178</a>
Bug Fix	Utility (Backup, Maintenance) jobs are failing when the log level is set to DEBUG.	<a href="#">#6208</a>

## 2.14 Version 8.0.0

New features, enhancements, bug fixes, and other changes in PEM 8.0.0 include:

Type	Description	ID
New Feature	<b>Security Best Practices:</b> <a href="#">Securing your deployment</a> documentation is available from our docs website. It will help you to setup PEM in a secure way to minimize the risk of vulnerabilities.	PEM-3802, PEM-3805
New Feature	<b>Webhooks:</b> You can now use PEM to send event based notifications to webhook endpoints like third party systems or partner applications.	PEM-3819
Enhancement	<b>Performance Diagnostics:</b> Use the Performance Diagnostic tool with PostgreSQL 10 or later versions after installing 'edb_wait_states' plugin.	NA



Type	Description	ID
Enhancement	<b>PEM Backend Database:</b> The PEM Server (backend database) supports only PostgreSQL or EDB Postgres Advanced Server versions 11 or later.	NA
Enhancement	<b>Enhanced Alerting:</b> You can now replace alert placeholders inside the script and also can view more informative and contextual alert details. [Support Ticket #1051538, #891377]	PEM-3612, PEM-3613, PEM-2501
Enhancement	<b>Enhanced BART Integration:</b> Configure 'bart_socket_name' and also parameters such as '--checksum-algorithm' and '--disable-checksum' through PEM.	PEM-3669
Enhancement	You can use the Macros feature in the Query tool.	<a href="#">#1402</a>
Enhancement	You can use the trigger function under the respective trigger node.	<a href="#">#2519</a>
Enhancement	You can download the utility files at the client side using Storage Manager.	<a href="#">#3318</a>
Enhancement	You can use the open Query tool tab to change the database connection.	<a href="#">#3794</a>
Enhancement	You can rename the Query tool and Debugger tab.	<a href="#">#4230</a>
Enhancement	You can ignore the owner while comparing the objects through Schema diff tool.	<a href="#">#5200</a>
Enhancement	Improved installation and upgrade guide as per customer feedback. [Support Ticket # 1101462]	PEM-3808
Bug Fix	Documented that while configuring the pem server, certificates must be present in data directory of backend database server in the installation guides. [Support Ticket # 729238]	PEM-672
Bug Fix	Connections not getting released when the user disconnects the database server. [Support Ticket #969833]	PEM-3184
Bug Fix	Fixed an issue where BART integration isn't working, when BART is installed at custom location. [Support Ticket # 1088574]	PEM-3737
Bug Fix	Do not include detailed alert information while sending SNMP traps. [Support Ticket #1069206, #1115277]	PEM-3791
Bug Fix	Two database connections made for failed login attempts which result in locking the user profile. [Support Ticket #1103288]	PEM-3816
Bug Fix	Package deployment and streaming replication deprecation warning is added in Upgrade and Migration Guide. [Support Ticket #1021617]	PEM-3799
Bug Fix	Ensure that some fields should be disabled for the trigger in edit mode.	<a href="#">#4639</a>
Bug Fix	Added a useful message when the explain plan is not used and empty.	<a href="#">#4806</a>
Bug Fix	File extension is stripped on renaming a file.	<a href="#">#4855</a>
Bug Fix	Ensure that the 'ctrl + a' shortcut does not move the cursor in the SQL Editor.	<a href="#">#5131</a>
Bug Fix	Fixed an issue where schema diff is showing identical tables as different due to default vacuum settings.	<a href="#">#5826</a>
Bug Fix	Fixed reverse engineering SQL where parenthesis is not properly arranged for View/MView definition.	<a href="#">#5830</a>
Bug Fix	Fixed 'can't execute an empty query' message if the user changes the option of Auto FK Index.	<a href="#">#5835</a>
Bug Fix	Fixed an issue where the server is not able to connect using the service.	<a href="#">#5841</a>
Bug Fix	Ensure that query history should be listed by date/time in descending order.	<a href="#">#5842</a>
Bug Fix	Fixed an issue where the 'PARALLEL UNSAFE' option is missing from reverse engineering SQL of function/procedure.	<a href="#">#5843</a>
Bug Fix	Fixed an issue where 'Rows X' column values were not visible properly for Explain Analyze in Dark theme.	<a href="#">#5853</a>
Bug Fix	Ensure that the user should be able to change the start value of the existing sequence.	<a href="#">#5855</a>
Bug Fix	Ensure that search object functionality works with case insensitive strings.	<a href="#">#5858</a>
Bug Fix	Fixed invalid literal issue when fetching dependencies for Materialized View.	<a href="#">#5882</a>
Bug Fix	Fixed an issue where the user is unable to change the macro name.	<a href="#">#5885</a>
Bug Fix	Fixed an issue where the suffix for Toast table size is not visible in the Statistics tab.	<a href="#">#5895</a>
Bug Fix	Ensure that macros should be run on the older version of Safari and Chrome.	<a href="#">#5911</a>
Bug Fix	Fixed an issue where a mismatch in the value of 'Estimated row' for functions.	<a href="#">#5914</a>
Bug Fix	Fixed an issue where non-closable tabs are getting closed.	<a href="#">#5923</a>
Bug Fix	Ensure that folder rename should work properly in Storage Manager.	<a href="#">#5943</a>
Bug Fix	Fixed an issue where a long file name is not visible on the process watcher dialog.	<a href="#">#5950</a>
Bug Fix	Fixed an issue where connection to the server is on wait state if a different user is provided.	<a href="#">#5953</a>
Bug Fix	Ensure that Grant Wizard should include foreign tables.	<a href="#">#5959</a>



Type	Description	ID
Bug Fix	Fixed an issue where the debugger's custom tab title did not apply when opened in the new browser tab.	<a href="#">#5974</a>
Bug Fix	Fixed an issue where the dynamic tab title has not applied the first time for the debugger panel.	<a href="#">#5978</a>
Bug Fix	Added the appropriate server icon based on the server type in the new connection dialog.	<a href="#">#5983</a>
Bug Fix	Fixed an issue where the process watcher dialog throws an error for the database server which is already removed.	<a href="#">#5985</a>

### 3 Platform compatibility

For information about the platforms and versions supported by PEM, see [Platform Compatibility](#).

#### Note

Postgres Enterprise Manager 8.3 and later is supported on SLES.

### 4 Hardware requirements for installing the PEM server on Linux

For optimum performance when monitoring servers and rendering dashboards, we recommend installing PEM on a system with at least:

- 4 CPU cores
- 8 GB of RAM
- 100 GB of storage

Additional disk space is required for data storage. Resource use varies based on the probes that are defined and enabled and the activity level on the monitored databases. Monitoring server resources as you use PEM lets you know when you need to expand your initial system configuration.

### 5 Understanding Postgres Enterprise Manager components and architecture

Postgres Enterprise Manager (PEM) monitors and manages multiple Postgres servers through a single graphical interface. PEM can monitor the following areas of the infrastructure:

- **Hosts** — One or more servers (physical or virtual) and their operating systems.
- **Database servers** — One or more instances of PostgreSQL or EDB Postgres Advanced Server or EDB Postgres Extended Server (formerly known as 2ndQPostgres) running on a host.
- **Databases** — One or more databases and their schema objects, such as tables and indexes.

#### Note

The term Postgres refers to PostgreSQL, EDB Postgres Advanced Server, or EDB Postgres Extended Server.

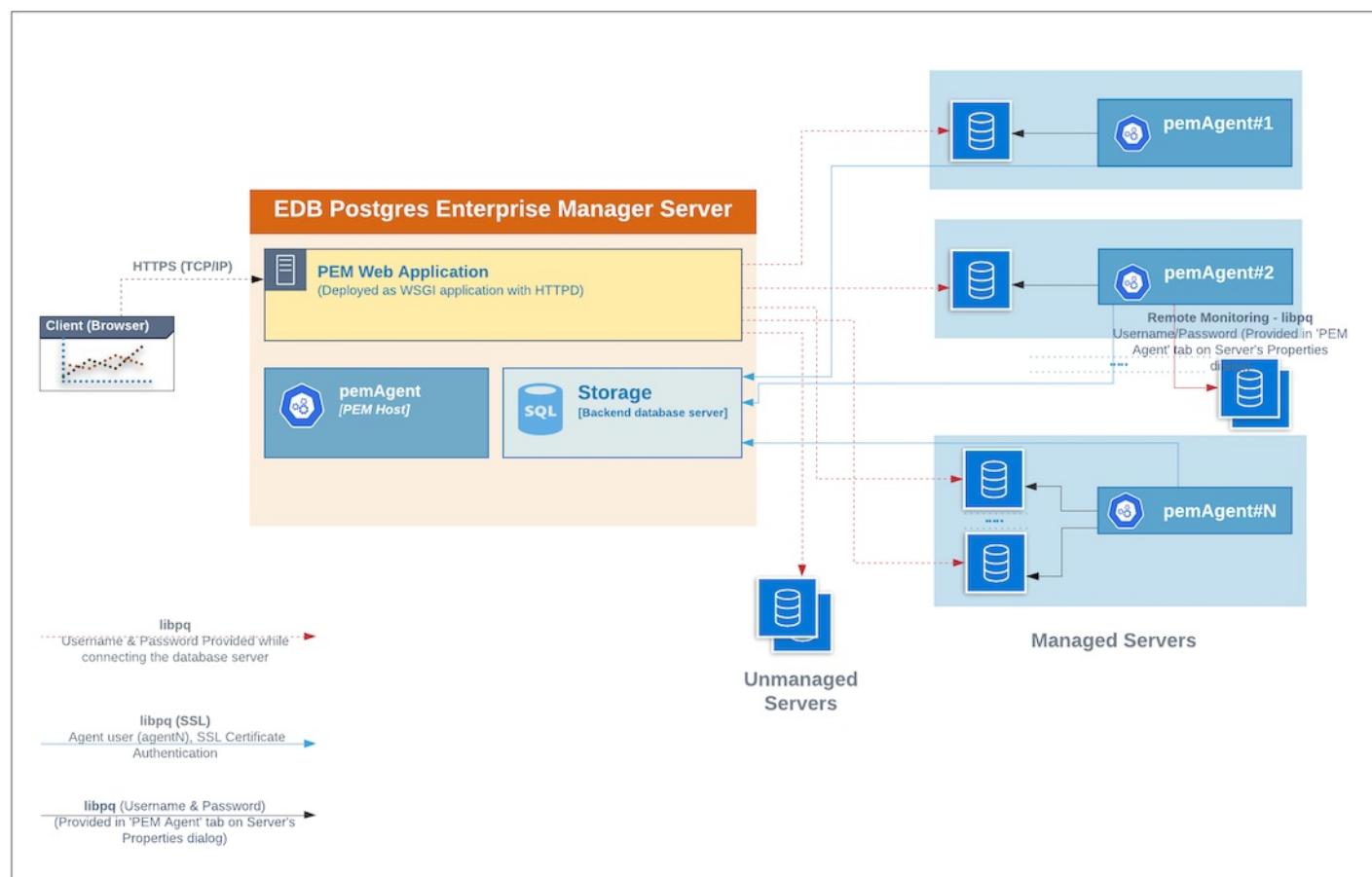
PEM consists of individual software components:

- **PEM server** — The PEM server is the data repository for monitoring data and a server to which agents and clients connect. The PEM server consists of an instance of PostgreSQL, an associated database for storing monitoring data, and a server that provides web services.
- **PEM agent** — The PEM agent is responsible for executing tasks and reporting statistics from the agent host and the monitored Postgres instances to the PEM server. A single PEM agent can monitor multiple installed instances of Postgres that reside on one or many hosts.
- **PEM web client** — The PEM web interface allows you to manage and monitor Postgres servers and use PEM extended functionality. The web interface software is installed with the PEM server and is accessed using any supported web browser.
- **SQL Profiler** — SQL Profiler is a Postgres server plugin to record the monitoring data and query plans for the SQL Profiler tool to analyze in PEM. This

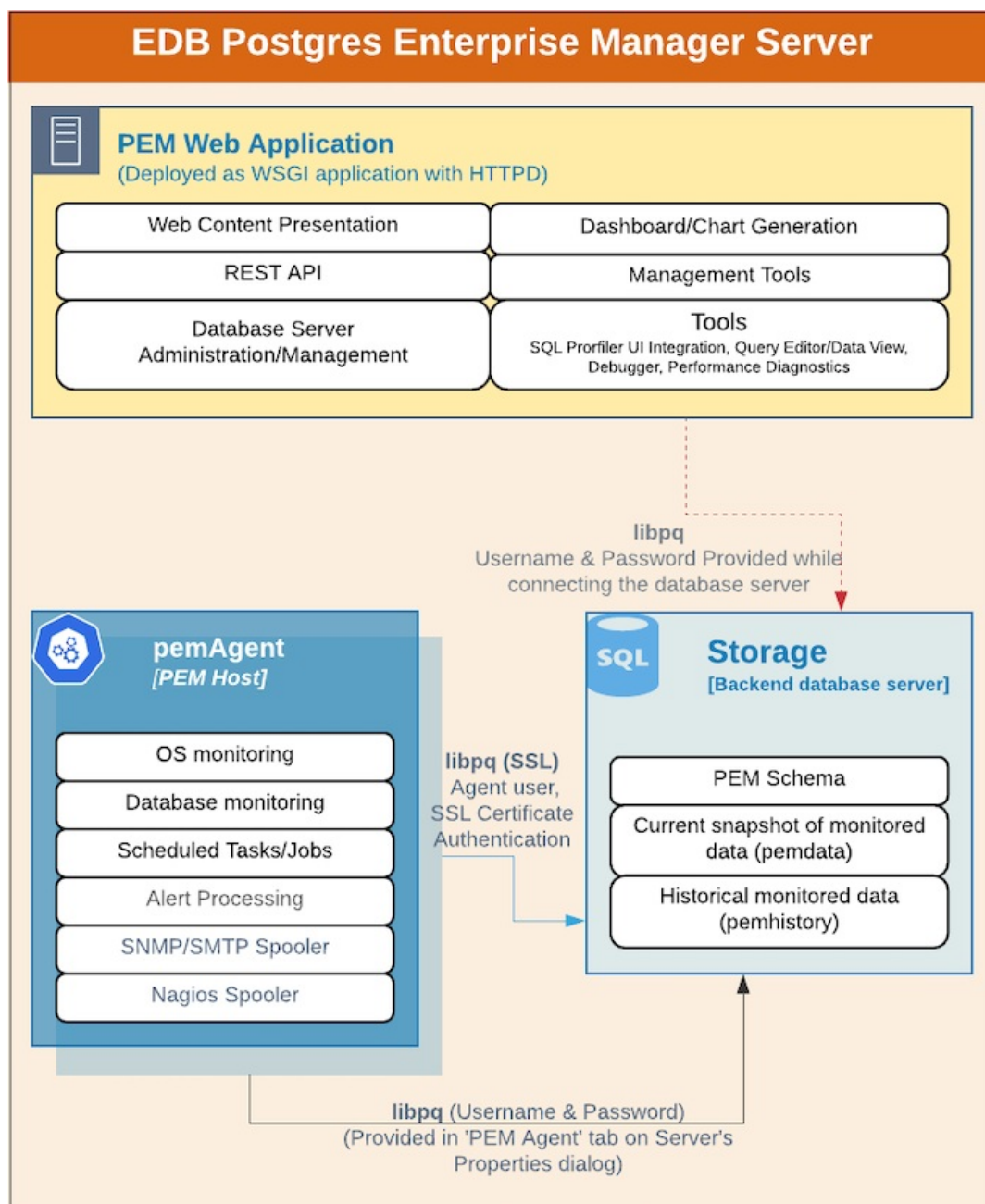
is an optional component of PEM, but the plugin must be installed in each instance of Postgres for which you want to use it. You can use the SQL Profiler with any supported version of an EDB distribution of a PostgreSQL server or EDB Postgres Advanced Server, not just those managed through the PEM server. See [SQL Profiler Configuration](#) for details and supported versions.

## PEM architecture

The following architectural diagram shows the relationships between the PEM server, clients, and managed and unmanaged Postgres servers.



## PEM server



The PEM server consists of an instance of Postgres, an instance of the Apache web server providing web services to the client, and a PEM Agent. PEM uses a server-side cryptographic plugin to generate authentication certificates.

The instance of Postgres (a database server) and an instance of the Apache web-server HTTPD can be on the same host or on separate hosts.

### !!! Note

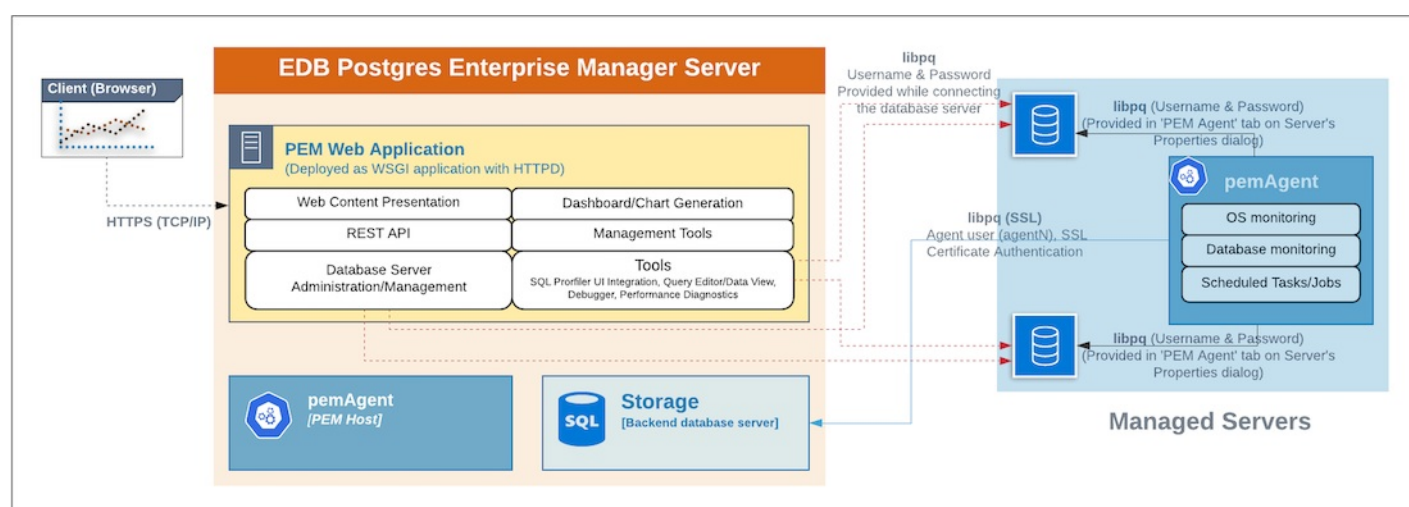
All the PEM features are available on either backend database server you select: PostgreSQL or EDB Postgres Advanced Server.

- **Postgres instance (database server)** — This is the backend database server. It hosts a database named `pem`, which acts as the repository for PEM server. The `pem` database contains several schemas that store metric data collected from each monitored host, server, and database.
  - `pem` — This schema is the core of the PEM application. It contains the definitions of configuration functions, tables, or views required by the application.
  - `pemdata` — This schema stores the current snapshot of the monitored data.
  - `pemhistory` — This schema stores the historical monitored data.
- **Apache web server (HTTPD)** — The PEM web application is deployed as a WSGI application with HTTPD to provide web services to the client. It is made up of the following:

- **Web content presentation** — The presentation layer is created by the web application (such as browser and login page).
- **Rest API** — The REST API allows integration with other apps and services.
- **Database server administration/management** — You can perform database server administration and management activities like CREATE, ALTER, and DROP for managed and unmanaged servers.
- **Dashboard/chart generation** — Internally, the web application includes functionality that generates dashboards and charts.
- **Management tools** — The Audit Manager, Capacity Manager, Log Manager, Postgres Expert, Postgres Log Analysis Expert, and the Tuning Wizard are available in the web application.
- Other tools provide functionality on managed or unmanaged servers:
  - **SQL Profiler UI integration** — SQL Profiler generates easily analyzed traces of session content.
  - **Query editor/data view** — The Query editor allows you to query, edit, and view data.
  - **Debugger** — The debugger helps you debug queries.
  - **Performance diagnostics** — Performance diagnostics help you analyze the performance of Postgres instances.

We recommend that you use a dedicated machine to host production instances of the PEM backend database. The host might be subject to high levels of data throughput, depending on the number of database servers that are being monitored and the workloads the servers are processing.

## PEM agent



The PEM agent is responsible for collecting monitoring data from the machine and operating system and from each of the Postgres instances to which they are bound. Each PEM agent can monitor one physical or virtual machine and is capable of monitoring multiple database servers locally that are installed on other systems. These servers can be installed on the same system or remotely. It's also responsible for executing other tasks that the user might schedule such as server shutdowns, SQL Profiler traces, and custom jobs.

A PEM agent is installed by default on the PEM server when you install the PEM server. It is generally referred to as a PEM agent on the PEM host. Separately, you can also install the PEM agent on the other servers hosting the Postgres instances you want to monitor using PEM.

Whether monitoring locally or remotely, the PEM agent connects to the PEM server using PostgreSQL's libpq, using SSL certificate-based authentication. The PEM agent installer in Windows and pemworker CLI in Linux is responsible for registering each agent with the PEM server and generating and installing the required certificates.

There is only one-way traffic between the PEM agent and PEM server. The PEM agent always connects to the PEM server.

The PEM agent must be able to connect to each database server that it monitors. This connection is made over a TCP/IP connection (or, optionally, a Unix Domain Socket on Unix hosts), and can optionally use SSL. You must configure the connection and authentication to the monitored server.

Once configured, each agent collects statistics and other information on the host and each database server and database that it monitors. Each piece of information is known as a *metric* and is collected by a *probe*. Most probes collect multiple metrics at once for efficiency. Examples of the metrics collected include:

- Disk I/O statistics
- Network statistics
- Database server version string
- Database server configuration option (GUC) values

- Table access statistics
- Table and index sizes

For a list of PEM probes, see [Probes](#).

By default, the PEM agent bound to the database server collects the OS/database monitoring statistics and also runs any scheduled tasks/jobs for that particular database server, storing data in the `pem` database on the PEM server.

The alert processing, SNMP/SMTP spoolers, and Nagios spooler data is stored in the `pem` database on the PEM server and is then processed by the PEM agent on the PEM host by default. However, you can enable processing by other PEM Agents by adjusting the SNMP/SMTP and Nagios parameters of the PEM agents.

For more information about these parameters, see [Server configuration](#).

**PEM web client**

The PEM client is a web-based application that runs in supported browsers. The client's web interface connects to the PEM server and allows direct management of managed or unmanaged servers and the databases and schemas that reside on them.

The client allows you to use PEM functionality that makes use of the data logged on the server through features such as dashboards, the Postgres Log Analysis Expert, and Capacity Manager.

**SQL Profiler plugin**

You don't have to install the SQL Profiler plugin on every server, but you must install and configure the plugin on each server on which you want to use the SQL Profiler. You might also want to install and configure SQL Profiler on unmonitored development servers. You can also temporarily install the SQL Profiler plugin for ad hoc use.

The plugin is installed with the EDB Postgres Advanced Server distribution but must be installed separately for use with PostgreSQL. The SQL Profiler installer is available from the [EDB website](#).

You can use SQL Profiler on servers that aren't managed through PEM. However, to perform scheduled traces, a server must have the plugin installed and must be managed by an installed and configured PEM agent.

For more information about using SQL Profiler, see [SQL Profiler](#).

**6 Deployment considerations**

There are a number of things to consider before deploying Postgres Enterprise Manager.

Considerations	Implementation instructions
Is a standalone server sufficient or do you need a high availability architecture?	<a href="#">Installing the server</a> or <a href="#">Deploying high availability</a>
Do you need to implement connection pooling?	<a href="#">Deploying connection pooling</a>
What type of authentication to use?	<a href="#">Authentication options</a>
What actions should you take to avoid security vulnerabilities?	<a href="#">Securing your deployment</a>
Where to host the Apache web server?	<a href="#">Web server installation options</a>

**6.1 Using Failover Manager for high availability**

Failover Manager is a high-availability tool from EDB that enables a Postgres primary node to failover to a standby node during a software or hardware failure on the primary.

You can set up Failover Manager only with a fresh installation of a Postgres Enterprise Manager (PEM) server. You can't set it for an existing PEM installation.

The examples in the following sections use these IP addresses:

- 172.16.161.200 - PEM Primary
- 172.16.161.201 - PEM Standby 1
- 172.16.161.202 - PEM Standby 2
- 172.16.161.203 - EFM Witness Node
- 172.16.161.245 - PEM VIP (used by agents and users to connect)

The following must use the VIP address:

- The PEM agent binding of the monitored database servers
- Accessing the PEM web client
- Accessing the webserver services

## Initial product installation and configuration

1. Install the following on the primary and one or more standbys:

- [EDB Postgres Advanced Server](#) (backend database for PEM Server)
- [PEM server](#)
- [EDB Failover Manager 4.1](#)

Refer to the installation instructions in the product documentation using these links or see the instructions on the [EDB repos website](#). Replace `USERNAME:PASSWORD` with your username and password in the instructions to access the EDB repositories.

Make sure that the database server is configured to use the scram-sha-256 authentication method, as the PEM server configuration script doesn't work with trust authentication.

You must install the `java-1.8.0-openjdk` package to install EFM.

2. Configure the PEM server on the primary server as well as on all the standby servers with an initial configuration of type 1 (web services and database):

```
/usr/edb/pem/bin/configure-pem-server.sh -t 1
```

For more detail on configuration types see, [Configuring the PEM server on Linux](#).

3. Add the following ports in the firewall on the primary and all the standby servers to allow the access:

- `8443` for PEM Server (https)
- `5444` for EPAS 13
- `7800` for EFM
- `7908` for EFM Admin

For example:

```
$ sudo firewall-cmd --zone=public --add-port=5444/tcp --permanent
success
$ sudo firewall-cmd --zone=public --add-port=8443/tcp --permanent
```

```

success
$ sudo firewall-cmd --zone=public --add-port=7800/tcp --permanent
success
$ sudo firewall-cmd --zone=public --add-port=7809/tcp --permanent
success
$ sudo firewall-cmd --reload
success

```

## Set up the primary node for streaming replication

1. Create the replication role:

```

$ /usr/edb/as13/bin/psql -h 172.16.161.200 -p 5444 -U enterprisedb edb -c "CREATE ROLE repl
REPLICATION LOGIN PASSWORD 'password'";
CREATE ROLE

```

Give the password of your choice.

2. Configure the following in the `postgresql.conf` file:

```

wal_level = replica
max_wal_senders = 10
wal_keep_size = 500
max_replication_slots =
10

```

For more information on configuring parameters for streaming replication, see the [PostgreSQL documentation](#).

### Note

The configuration parameters might differ for different versions of the database server. You can email EDB Support at [techsupport@enterprisedb.com](mailto:techsupport@enterprisedb.com) for help with setting up these parameters.

3. Add the following entry in the host-based authentication (`/var/lib/edb/as13/data/pg_hba.conf`) file to allow the replication user to connect from all the standbys:

```

hostssl replication repl 172.16.161.201/24 scram-sha-256

```

### Note

You can change the cidr range of the IP address, if needed.

4. Modify the host-based authentication (`/var/lib/edb/as13/data/pg_hba.conf`) file for the `pem_user` role to connect to all databases using the `scram-sha-256` authentication method:

```

# Allow local PEM agents and admins to connect to PEM server
hostssl all +pem_user 172.16.161.201/24 scram-sha-256
hostssl pem +pem_user 127.0.0.1/32 scram-sha-256
hostssl pem +pem_agent 127.0.0.1/32 cert
# Allow remote PEM agents and users to connect to PEM server
hostssl pem +pem_user 0.0.0.0/0 scram-sha-256
hostssl pem +pem_agent 0.0.0.0/0 cert

```

5. Restart the EPAS 13 server.

```
systemctl restart edb-as-13.service
```

## Set up the standby nodes for streaming replication

1. Stop the service for EPAS 13 on all the standby nodes:

```
$ systemctl stop edb-as-13.service
```

### Note

This example uses the `pg_basebackup` utility to create the replicas of the PEM backend database server on the standby servers. When using `pg_basebackup`, you need to stop the existing database server and remove the existing data directories.

2. Remove the data directory of the database server on all the standby nodes:

```
$ sudo su - enterprisedb

$ rm -rf /var/lib/edb/as13/data/*
```

3. Create the `.pgpass` file in the home directory of the `enterprisedb` user on all the standby nodes:

```
$ sudo su - enterprisedb

$ cat > ~/.pgpass << _EOF_
172.16.161.200:5444:replication:repl:CHANGE_ME
172.16.161.201:5444:replication:repl:CHANGE_ME
172.16.161.202:5444:replication:repl:CHANGE_ME
_EOF_

$ chmod 600 ~/.pgpass
```

4. Take the backup of the primary node on each of the standby nodes using `pg_basebackup`:

```
$ sudo su - enterprisedb /usr/edb/as13/bin/pg_basebackup -h 172.16.161.200 \
-D /var/lib/edb/as13/data -U repl -v -P -Fp -R -p 5444
```

The `backup` command creates the `postgresql.auto.conf` and `standby.signal` files on the standby nodes. The `postgresql.auto.conf` file has the following content:

```
sudo su - enterprisedb cat /var/lib/edb/as13/data/postgresql.auto.conf
# Do not edit this file manually
# It will be overwritten by the ALTER SYSTEM command.
primary_conninfo = 'user=repl passfile=''/var/lib/edb/.pgpass'' channel_binding=prefer
host=172.16.161.200 port=5444 sslmode=prefer sslcompression=0 ssl_min_protocol_version=TLSv1.2
gssencmode=prefer krbsvrname=postgres target_session_attrs=any'
```

5. In the `postgresql.conf` file on each of the standby nodes, edit the following parameter:

```
hot_standby =
on
```

6. Start the EPAS 13 database server on each of the standby nodes:



```
$ systemctl enable edb-as-13
```

```
$ systemctl start edb-as-13
```

7. Copy the following files from the primary node to the standby nodes at the same location, overwriting any existing files. Set the permissions on the files:

- o `/etc/httpd/conf.d/edb-pem.conf`
- o `/etc/httpd/conf.d/edb-ssl-pem.conf`
- o `/root/.pem/agent1.crt`
- o `/root/.pem/agent1.key`
- o `/usr/edb/pem/agent/etc/agent.cfg`
- o `/usr/edb/pem/share/.install-config`
- o `/usr/edb/pem/web/pem.wsgi`
- o `/usr/edb/pem/web/config_setup.py`

For example:

```
$ mkdir -p /root/.pem
$ chown root:root /root/.pem
$ chmod 0755 /root/.pem
$ mkdir -p /var/lib/pemhome/.pem
$ chown pem:pem /var/lib/pemhome/.pem
$ chmod 0700 /var/lib/pemhome/.pem
$ mkdir -p /usr/edb/pem/logs
$ chown root:root /usr/edb/pem/logs
$ chmod 0755 /usr/edb/pem/logs
$ for file in /etc/httpd/conf.d/edb-pem.conf \
              /etc/httpd/conf.d/edb-ssl-pem.conf \
              /root/.pem/agent1.crt \
              /usr/edb/pem/agent/etc/agent.cfg \
              /usr/edb/pem/share/.install-config \
              /usr/edb/pem/web/pem.wsgi \
              /usr/edb/pem/web/config_setup.py; do \
    chown root:root ${file}; \
    chmod 0644 ${file}; \
done;
$ chmod 0600 /root/.pem/agent1.key
$ chown root:root /root/.pem/agent1.key
```

This code ensures that the webserver is configured on the standby and is disabled by default. Switchover by EFM enables the webserver.

#### Note

Manually keep the certificates in sync on master and standbys whenever the certificates are updated.

8. Run the `configure-selinux.sh` script to configure the SELinux policy for PEM:

```
$ /usr/edb/pem/bin/configure-selinux.sh
getenforce found, now executing 'getenforce' command
Configure the httpd to work with the SELinux
Allow the httpd to connect the database (httpd_can_network_connect_db = on)
Allow the httpd to connect the network (httpd_can_network_connect = on)
Allow the httpd to work with cgi (httpd_enable_cgi = on)
Allow to read & write permission on the 'pem' user home directory
SELinux policy is configured for PEM

$ sudo chmod 640 /root/.pem/agent1.crt
```

9. Disable and stop HTTPD and PEM agent services if they're running on all replica nodes:

```
systemctl stop pemagent
systemctl stop httpd
systemctl disable pemagent
systemctl disable httpd
```

#### Note

At this point, a PEM primary server and two standbys are ready to take over from the primary whenever needed.

## Set up EFM to manage failover on all hosts

1. Prepare the primary node to support EFM:

- Create a database user efm to connect to the database servers.
- Grant the execute privileges on the functions related to WAL logs and the monitoring privileges to the user.
- Add entries in `pg_hba.conf` to allow the efm database user to connect to the database server from all nodes on all the hosts.
- Reload the configurations on all the database servers.

For example:

```
$ cat > /tmp/efm-role.sql <<
_EOF_
-- Create a role for
EFM
CREATE ROLE efm LOGIN PASSWORD
'password';

-- Give privilege to 'efm' user to connect to a
database
GRANT CONNECT ON DATABASE edb TO
efm;

-- Give privilege to 'efm' user to do backup
operations
GRANT EXECUTE ON FUNCTION pg_current_wal_lsn() TO efm;
GRANT EXECUTE ON FUNCTION pg_last_wal_replay_lsn() TO
efm;
GRANT EXECUTE ON FUNCTION pg_wal_replay_resume() TO efm;
GRANT EXECUTE ON FUNCTION pg_wal_replay_pause() TO
efm;
GRANT EXECUTE ON FUNCTION pg_reload_conf() TO
efm;

-- Grant monitoring privilege to the 'efm'
user
GRANT pg_monitor TO efm;
_EOF_

$ /usr/edb/as13/bin/psql -h 172.16.161.200 -p 5444 -U enterprisedb edb -f
/tmp/efm-role.sql
CREATE ROLE
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT ROLE
```

```

$ rm -f
/tmp/efm-role.sql

$ cat > /var/lib/edb/as13/data/pg_hba.conf <<<
_EOF_
    hostssl            edb        efm        172.16.161.200/32
    scram-sha-256
    hostssl            edb        efm        172.16.161.201/32
    scram-sha-256
    hostssl            edb        efm        172.16.161.202/32
    scram-sha-256
    hostssl            edb        efm        172.16.161.203/32
    scram-sha-256
_EOF_

$ /usr/edb/as13/bin/psql -h 172.16.161.200 -p 5444 -U enterprisedb edb -c "SELECT
pg_reload_conf();"

```

2. Create the scripts on each node to start/stop the PEM agent:

```

$ sudo cat > /usr/local/bin/start-httpd-pemagent.sh << _EOF_
#!/bin/sh
/bin/sudo /bin/systemctl enable httpd
/bin/sudo /bin/systemctl start httpd
/bin/sudo /bin/systemctl enable pemagent
/bin/sudo /bin/systemctl start pemagent
_EOF_
$ sudo cat > /usr/local/bin/stop-httpd-pemagent.sh << _EOF_
#!/bin/sh

/bin/sudo /bin/systemctl stop pemagent
/bin/sudo /bin/systemctl disable pemagent
/bin/sudo /bin/systemctl stop httpd
/bin/sudo /bin/systemctl disable httpd
_EOF_
$ sudo chmod 770 /usr/local/bin/start-pemagent.sh
$ sudo chmod 770 /usr/local/bin/stop-pemagent.sh

```

3. Create a `sudoers` file ( `/etc/sudoers.d/efm-pem` ) on each node to allow the efm user to start/stop the pemagent:

```

$ sudo cat > /etc/sudoers.d/efm-pem << _EOF_
efm    ALL=(ALL)          NOPASSWD:    /bin/systemctl enable pemagent
efm    ALL=(ALL)          NOPASSWD:    /bin/systemctl disable pemagent
efm    ALL=(ALL)          NOPASSWD:    /bin/systemctl stop pemagent
efm    ALL=(ALL)          NOPASSWD:    /bin/systemctl start pemagent
efm    ALL=(ALL)          NOPASSWD:    /bin/systemctl status pemagent
_EOF_

```

4. Create an `efm.nodes` file on all nodes using the sample file ( `/etc/edb/efm-4.1/efm.nodes.in` ), and give read-write access to the efm OS user:

```

$ sudo cp /etc/edb/efm-4.1/efm.nodes.in /etc/edb/efm-4.1/efm.nodes
$ sudo chown efm:efm /etc/edb/efm-4.1/efm.nodes
$ sudo chmod 600 /etc/edb/efm-4.1/efm.nodes

```

5. Add the IP address and efm port of the primary node in the `/etc/edb/efm-4.1/efm.nodes` file on the standby nodes:

```

$ sudo cat > /etc/edb/efm-4.1/efm.nodes <<< _EOF_
172.16.161.200:7800

```

`_EOF_`

6. Create the `efm.properties` file on all the nodes using the sample file (`/etc/edb/efm-4.1/efm.properties.in`). Grant read access to all the users:

```
$ sudo cp /etc/edb/efm-4.1/efm.properties.in /etc/edb/efm-4.1/efm.properties
$ sudo chown efm:efm /etc/edb/efm-4.1/efm.properties
$ sudo chmod a+r /etc/edb/efm-4.1/efm.properties
```

7. Encrypt the efm user's password using the efm utility:

```
$ export EFMPASS=password
$ /usr/edb/efm-4.1/bin/efm encrypt efm --from-env
096666746b05b081d1a98e43d94c9dad
```

8. Edit the following parameters in the properties file:

```
db.user=efm

db.password.encrypted=096666746b05b081d1a98e43d94c9dad
db.port=5444

db.database=edb

db.service.owner=enterprisedb
db.service.name=edb-as-13

db.bin=/usr/edb/as13/bin

db.data.dir=/var/lib/edb/as13/data
jdbc.sslmode=require
user.email=username@example.com
from.email=node1@efm-pem
notification.level=INFO
notification.text.prefix=[PEM/EFM]
bind.address=172.16.161.200:7800
admin.port=7809
is.witness=false
local.period=10
local.timeout=60

local.timeout.final=10

remote.timeout=10
node.timeout=50
encrypt.agent.messages=true

stop.isolated.primary=true

stop.failed.primary=true
primary.shutdown.as.failure=false
update.physical.slots.period=0

ping.server.ip=8.8.8.8
ping.server.command=/bin/ping -q -c3 -w5

auto.allow.hosts=false
stable.nodes.file=false
db.reuse.connection.count=0
auto.failover=true
```

```

auto.reconfigure=true
  promotable=true

use.replay.tiebreaker=true

standby.restart.delay=0
  reconfigure.num.sync=false

reconfigure.sync.primary=false

minimum.standbys=0

recovery.check.period=1

restart.connection.timeout=60
  auto.resume.period=0
  virtual.ip=172.16.161.245
  virtual.ip.interface=ens33
  virtual.ip.prefix=24
  virtual.ip.single=true

check.vip.before.promotion=true
  pgpool.enable=false
  sudo.command=sudo
  sudo.user.command=sudo -u
%u

syslog.host=localhost

syslog.port=514
  syslog.protocol=UDP
  syslog.facility=LOCAL1

file.log.enabled=true

syslog.enabled=false

jgroups.loglevel=INFO
  efm.loglevel=INFO
  jvm.options=-
Xmx128m
  script.remote.post.promotion=/usr/local/bin/stop-pemagent.sh
  script.post.promotion=/usr/local/bin/start-
pemagent.sh

```

9. Set the value of the `is.witness` configuration parameter on the witness node to `true` :

```
is.witness=true
```

10. Enable and start the EFM service on the primary node:

```

$ systemctl enable edb-efm-4.1
$ systemctl start edb-efm-4.1

```

11. Allow the standbys to join the cluster started on the primary node:

```

/usr/edb/efm-4.1/bin/efm allow-node efm 172.16.161.201
/usr/edb/efm-4.1/bin/efm allow-node efm 172.16.161.202
/usr/edb/efm-4.1/bin/efm allow-node efm 172.16.161.203

```

12. Enable and start the EFM service on the standby nodes and the EFM witness node:

```
$ systemctl enable edb-efm-4.1
$ systemctl start edb-efm-4.1
```

13. Check the EFM cluster status from any node:

```
$ sudo /usr/edb/efm-4.1/bin/efm cluster-status efm
Cluster Status: efm
Agent Type  Address          DB      VIP
-----
Primary     172.16.161.200   UP      172.16.161.245*
Standby     172.16.161.201   UP      172.16.161.245
Standby     172.16.161.202   UP      172.16.161.245
Witness     172.16.161.203   N/A     172.16.161.245

Allowed node host list:
172.16.161.200 172.16.161.201 172.16.161.202 172.16.161.203

Membership coordinator: 172.16.161.200

Standby priority host list:
172.16.161.201 172.16.161.202

Promote Status:

DB Type      Address          WAL Received LSN  WAL Replayed LSN  Info
-----
Primary      172.16.161.200               0/F7A3808
Standby      172.16.161.201   0/F7A3808         0/F7A3808
Standby      172.16.161.202   0/F7A3808         0/F7A3808

Standby database(s) in sync with primary. It is safe to promote.
```

This status confirms that EFM is set up successfully and managing the failover for the PEM server.

In case of failover, any of the standbys are promoted as the primary node, and PEM agents connect to the new primary node. You can replace the failed primary node with a new standby using this procedure.

## Current limitations

The current limitations include:

- Web console sessions for the users are lost during the switchover.
- Per-user settings set from the Preferences dialog box are lost, as they're stored in local configuration files on the file system.
- Background processes, started by the Backup, Restore, and Maintenance dialogs boxes, and their logs aren't shared between the systems. They are lost during switchover.

## 6.2 Connection pooling using pgBouncer

You can use pgBouncer as a connection pooler for limiting the number of connections from the PEM agent to the Postgres Enterprise Manager (PEM) server on non-Windows machine:

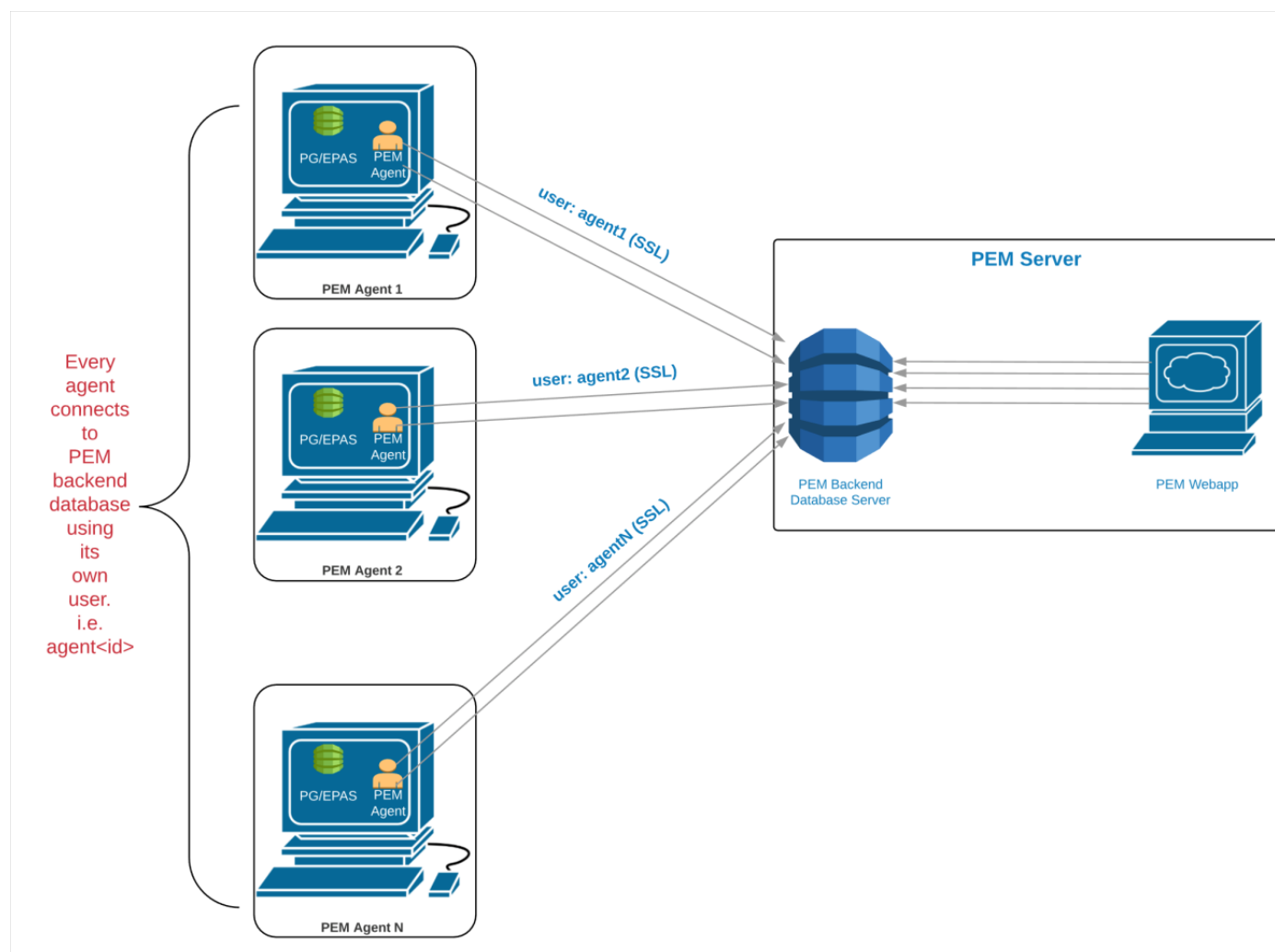
- Preparing the PEM Database Server provides information about preparing the PEM database server to be used with pgBouncer.
- Configuring pgBouncer provides detailed information about configuring pgBouncer to make it work with the PEM database server.

- Configuring the PEM agent provides detailed information about configuring a PEM agent to connect to pgBouncer.

For detailed information about using the PEM web interface, see the [Accessing the web interface](#).

## 6.2.1 PEM server and agent connection management mechanism

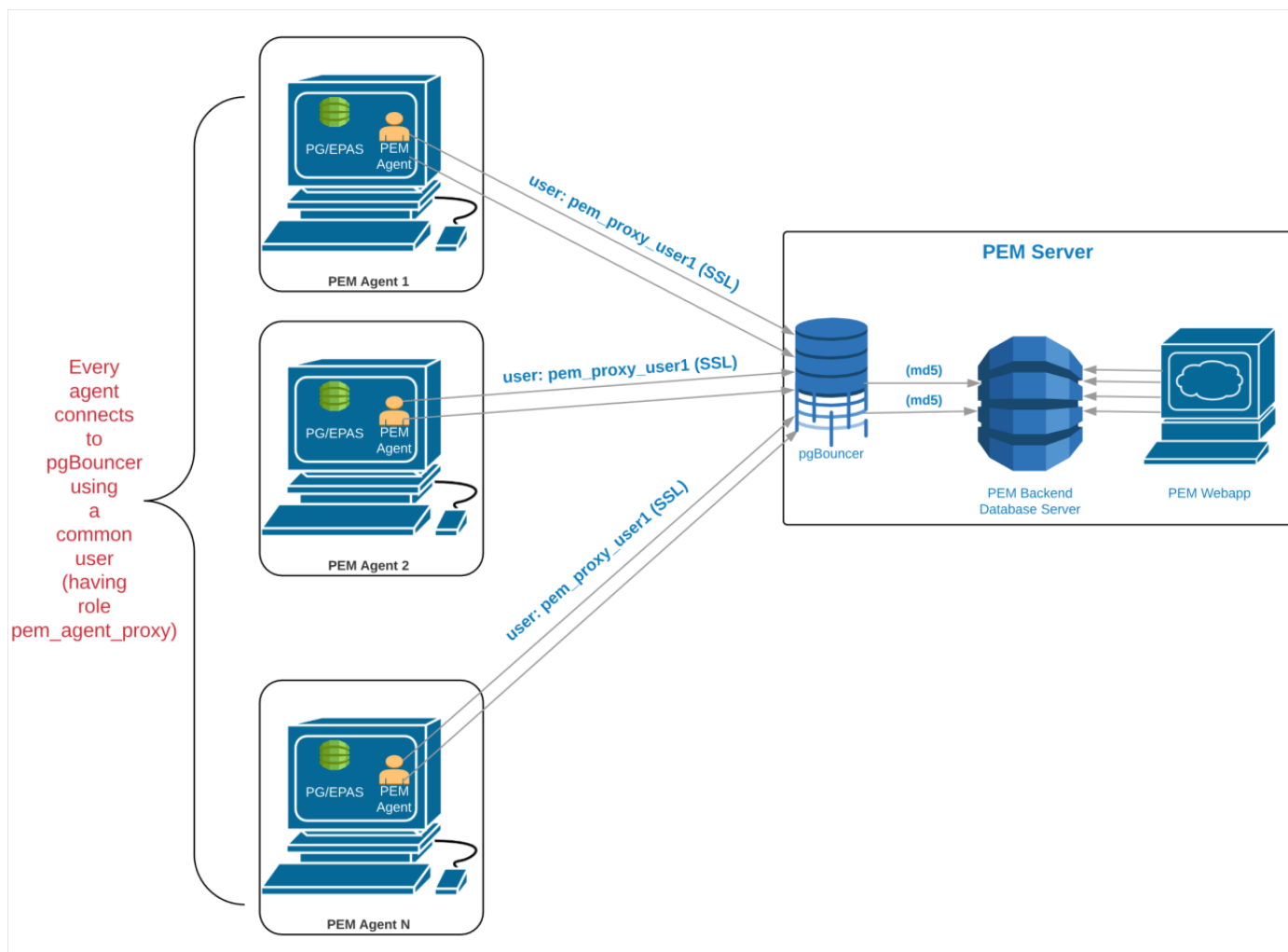
Each PEM agent connects to the PEM database server using the SSL certificates for each user. For example, an agent with `ID#1` connects to the PEM database server using the `agent1` user.



Prior to PEM version 7.5, the following limitations disallowed the use of the connection pooler between the PEM server and PEM agent:

- The PEM agent uses an SSL certificate to connect to the PEM database server.
- It uses an individual user identifier when connecting to the PEM database server.

EDB modified the PEM agent to allow the agent to use a common database user (instead of the dedicated agent users) to connect to the PEM database server.



We recommend using PgBouncer version 1.9.0 or later as the connection pooler. Versions 1.9.0 or later support cert authentication. PEM agents can connect to pgBouncer using SSL certificates.

## 6.2.2 Preparing the PEM database server

You must configure the PEM database server to work with PgBouncer. This example shows how to configure the PEM database server.

1. Create a dedicated user named pgbouncer on the PEM database server:

```
CREATE USER pgbouncer PASSWORD 'ANY_PASSWORD'
LOGIN;
```

```
CREATE ROLE
```

2. Create a user named pem\_admin1 (not a superuser) with pem\_admin and pem\_agent\_pool role membership on the PEM database server:

```
CREATE USER pem_admin1 PASSWORD 'ANY_PASSWORD' LOGIN
CREATEROLE;
```

```
CREATE ROLE
```

```
GRANT pem_admin, pem_agent_pool TO
pem_admin1;
```



```
GRANT ROLE
```

- Grant CONNECT privileges to the pgbouncer user on the `pem` database:

```
GRANT CONNECT ON DATABASE pem TO
pgbouncer;
```

```
GRANT
```

- Grant USAGE privileges to the pgbouncer user for the `pem` schema on the `pem` database:

```
GRANT USAGE ON SCHEMA pem TO
pgbouncer;
```

```
GRANT
```

- Grant EXECUTE privileges to the pgbouncer user on the `pem.get_agent_pool_auth(text)` function in the `pem` database. For example:

```
GRANT EXECUTE ON FUNCTION pem.get_agent_pool_auth(text) TO
pgbouncer;
```

```
GRANT
```

- Use the `pem.create_proxy_agent_user(varchar)` function to create a user named `pem_agent_user1` on the PEM database server:

```
SELECT pem.create_proxy_agent_user('pem_agent_user1');
```

```
create_proxy_agent_user
-----
(1 row)
```

The function creates a user with the same name with a random password and grants `pem_agent` and `pem_agent_pool` roles to the user. This approach allows pgBouncer to use a proxy user on behalf of the agent.

- Add the following entries to the start of the `pg_hba.conf` file of the PEM database server. These entries allow the pgBouncer user to connect to the `pem` database using the md5 authentication method.

```
# Allow the PEM agent proxy user (used by
# pgbouncer) to connect the to PEM server using
# md5

local pem pgbouncer,pem_admin1 md5
```

## 6.2.3 Configuring PgBouncer

You must configure PgBouncer to work with the PEM database server. This example runs PgBouncer as the `enterprisedb` system user and outlines the process of configuring pgBouncer version 1.9 or later.

- Open a terminal window and navigate to the pgBouncer directory.
- Change the owner of the `etc` directory for pgBouncer (where `pgbouncer.ini` resides) to `enterprisedb`, and change the directory permissions to `0700`:

```
$ chown enterprisedb:enterprisedb /etc/edb/pgbouncer1.9
$ chmod 0700 /etc/edb/pgbouncer1.9
```

3. Change the contents of the `pgbouncer.ini` or `edb-pgbouncer.ini` file:

```
[databases]
;; Change the pool_size according to maximum connections
allowed
;; to the PEM database server as
required.
;; 'auth_user' will be used for authenticate the db user
(proxy
;; agent user in our
case)

pem = port=5444 host=/tmp dbname=pem
auth_user=pgbouncer
pool_size=80
pool_mode=transaction
* = port=5444 host=/tmp dbname=pem
auth_user=pgbouncer
pool_size=10

[pgbouncer]
logfile = /var/log/edb/pgbouncer1.9/edb-pgbouncer-1.9.log
pidfile = /var/run/edb/pgbouncer1.9/edb-pgbouncer-1.9.pid
listen_addr =
*
;; Agent needs to use this port to connect the pem database
now
listen_port = 6432
;; Require to support for the SSL Certificate
authentications
;; for PEM
Agents
client_tls_sslmode = require
;; These are the root.crt, server.key, server.crt files
present
;; in the present under the data directory of the PEM
database
;; server, used by the PEM Agents for
connections.
client_tls_ca_file =
/var/lib/edb/as11/data/root.crt
client_tls_key_file = /var/lib/edb/as11/data/server.key
client_tls_cert_file = /var/lib/edb/as11/data/server.crt
;; Use hba file for client
connections
auth_type =
hba
;; Authentication file,
Reference:
;;
https://pgbouncer.github.io/config.html#auth\_file
auth_file =
/etc/edb/pgbouncer1.9/userlist.txt
;; HBA
file
auth_hba_file = /etc/edb/pgbouncer1.9/hba_file
;; Use pem.get_agent_pool_auth(TEXT) function to
authenticate
;; the db user (used as a proxy agent
user).
auth_query = SELECT * FROM
pem.get_agent_pool_auth($1)
```

```

;; DB User for administration of the
pgbouncer
admin_users = pem_admin1
;; DB User for collecting the statistics of
pgbouncer
stats_users = pem_admin1
server_reset_query = DISCARD
ALL
;; Change based on the number of agents
installed/required
max_client_conn = 500
;; Close server connection if its not been used in this
time.
;; Allows to clean unnecessary connections from pool after
peak.
server_idle_timeout = 60

```

4. Create and update the `/etc/edb/pgbouncer1.9/userlist.txt` authentication file for PgBouncer:

```

pem=# COPY
(
SELECT 'pgbouncer'::TEXT, 'pgbouncer_password'
UNION ALL
SELECT 'pem_admin1'::TEXT, 'pem_admin1_password'
TO '/etc/edb/pgbouncer1.9/userlist.txt'
WITH (FORMAT CSV, DELIMITER ' ', FORCE_QUOTE
*);
COPY 2

```

#### Note

A superuser can't invoke the PEM authentication query function `pem.get_proxy_auth(text)`. If the `pem_admin` user is a superuser, you must add the password to the authentication file (`enterprisedb` in the example).

5. Create an HBA file (`/etc/edb/pgbouncer1.9/hba_file`) for PgBouncer that contains the following content:

```

# Use authentication method md5 for the local connections to
# connect pem database & pgbouncer (virtual) database.
local pgbouncer all md5
# Use authentication method md5 for the remote connections to
# connect to pgbouncer (virtual database) using enterprisedb
# user.

host pgbouncer,pem pem_admin1 0.0.0.0/0 md5

```

```

# Use authentication method cert for the TCP/IP connections to
# connect the pem database using pem_agent_user1

hostssl pem pem_agent_user1 0.0.0.0/0 cert

```

6. Change the owner of the HBA file (`/etc/edb/pgbouncer1.9/hba_file`) to `enterprisedb`, and change the directory permissions to `0600`:

```

$ chown enterprisedb:enterprisedb /etc/edb/pgbouncer1.9/hba_file
$ chmod 0600 /etc/edb/pgbouncer1.9/hba_file

```

7. Enable the PgBouncer service, and start the service:

```
$ systemctl enable edb-pgbouncer-1.9

Created symlink from
/etc/systemd/system/multi-user.target.wants/edb-pgbouncer-1.9.service
to /usr/lib/systemd/system/edb-pgbouncer-1.9.service.

$ systemctl start edb-pgbouncer-1.9
```

## 6.2.4 Configuring the PEM agent

You can use an RPM package to install a PEM agent. For detailed installation information, see [Installing agent on Linux](#).

Don't configure the PEM agent responsible for sending SNMP notifications with pgBouncer. For example, if the default PEM agent installed with PEM server is used for SNMP notifications, don't configure it with pgBouncer.

### Configuring a new PEM agent (installed via RPM)

After using an RPM package to install the PEM agent, you must configure it to work against a particular PEM database server. Use the following command:

```
$ PGSSLMODE=require PEM_SERVER_PASSWORD=pem_admin1_password
/usr/edb/pem/agent/bin/pemworker --register-agent --pem-server
pem_agent_user1 --display-name "Agent Name"
```

Postgres Enterprise Manager Agent registered successfully!

In this command, the `--pem-agent-user` argument instructs the agent to create an SSL certificate and key pair for the `pem_agent_user1` database user in the `/root/.pem` directory.

For example:

```
/root/.pem/pem_agent_user1.crt
/root/.pem/pem_agent_user1.key
```

The PEM agent uses the keys to connect to the PEM database server as `pem_agent_user1`. It also creates an agent configuration file named `/usr/edb/pem/agent/etc/agent.cfg`.

A line mentioning the agent-user to use appears in the `agent.cfg` configuration file. For example:

```
$ cat
/usr/edb/pem/agent/etc/agent.cfg
[PEM/agent]
pem_host=172.16.254.22
pem_port=6432
agent_id=12
agent_user=pem_agent_user1
agent_ssl_key=/root/.pem/pem_agent_user1.key
agent_ssl_cert=/root/.pem/pem_agent_user1.crt
log_level=warning
log_location=/var/log/pem/worker.log
agent_log_location=/var/log/pem/agent.log
```

```

long_wait=30
short_wait=10
alert_threads=0
enable_smtp=false
enable_snmp=false
enable_webhook=false
max_webhook_retries=3
allow_server_restart=true
allow_package_management=false
allow_streaming_replication=false
max_connections=0
connect_timeout=-1
connection_lifetime=0
allow_batch_probes=false
heartbeat_connection=false

```

### Configuring an existing PEM agent (installed via RPM)

If you're using an existing PEM agent, you can copy the SSL certificate and key files to the target machine and reuse the files. You must modify the files, adding a new parameter and replacing some parameters in the existing `agent.cfg` file.

Add a line to use `agent_user` as the agent:

```
agent_user=pem_agent_user1
```

Update the port to specify the pgBouncer port:

```
pem_port=6432
```

Update the certificate and key path locations:

```

agent_ssl_key=/root/.pem/pem_agent_user1.key
agent_ssl crt=/root/.pem/pem_agent_user1.crt

```

As an alternative, you can run the agent self-registration script. However, that process creates a new agent id. If run the agent self-registration script, you must replace the new agent id with the existing id and disable the entry for the new agent id in the `pem.agent` table. For example:

```

pem=# UPDATE pem.agent SET active = false WHERE id =
<new_agent_id>;

```

```
UPDATE 1
```

#### Note

Keep a backup of the existing SSL certificate, key file, and agent configuration file.

## 6.3 Authentication options

PEM also supports Kerberos and 2FA authentication. For implementation instructions, see:

On Linux:

- [Configuring the PEM server to use Kerberos authentication](#)
- [Configuring the PEM server to use Windows Kerberos server](#)

On Linux and Windows:

- [Configuring the PEM server to use two-factor authentication](#)

### 6.3.1 Configuring the PEM server to use Kerberos authentication

**\*\*New Feature\*\***

Kerberos support is available in PEM 8.1.0 and later.

You can configure Kerberos authentication for the PEM server. The Kerberos server works with hostnames and not with IP addresses. To use single sign-on in PEM server using Kerberos authentication, configure the following machines with hostnames using the DNS (realm).

- Kerberos server
- PEM server (PEM web server and PEM backend database server)
- Client machine

For example, if the realm on Kerberos server is `edbpem.org`, then you can set the Kerberos server hostname to `Krb5server.edbpem.org`, the PEM server hostname to `pem.edbpem.org`, and the client's hostname to `pg12.edbpem.org`. The convention is to use the DNS domain name as the name of the realm.

#### 1. Install Kerberos, the PEM server, and the PEM backend database

Install Kerberos on the machine that functions as the authentication server. Install the PEM server on a separate machine. For more information, see [Installing the PEM Server](#).

Install the PEM backend database (Postgres/EDB Postgres Advanced Server) on the same machine as the PEM server or on a different one. For more information, see the Installation steps on [EDB Docs website](#).

#### 2. Add principals on Kerberos server

Add the principals for the PEM web application deployed under an Apache web server (HTTPD/Apache2) and the PEM Backend Database Server (PostgreSQL/EDB Postgres Advanced Server).

```
$ sudo kadmin.local -q "addprinc -randkey HTTP/<HOSTNAME_OF_PEM_SERVER>"
$ sudo kadmin.local -q "addprinc -randkey postgres/<HOSTNAME_OF_PEM_SERVER>"
```

`HOSTNAME_OF_PEM_SERVER` must contain the realm of the Kerberos server. For example, you can specify `pemdb.edbpem.org` as the hostname of PEM server, with `edbpem.org` as the realm.

#### Note

If the PEM web application and the PEM backend database server are on different machines, then hostname is different for each one.

### 3. Extract key tables from Kerberos server

Extract the key tables from Kerberos for the PEM web application and the PEM backend database server:

```
sudo kadmin.local "ktadd -k <NAME_OF_PEM_WEB_FILE>.keytab HTTP/<HOSTNAME_OF_PEM_SERVER>"
sudo kadmin.local "ktadd -k <NAME_OF_PEM_DB_FILE>.keytab postgres/<HOSTNAME_OF_PEM_SERVER>"
```

Copy the key tables from the Kerberos server to the PEM server:

```
scp <NAME_OF_PEM_WEB_FILE>.keytab <OS_USERNAME_ON_PEM_SERVER>@<HOSTNAME_OF_PEM_SERVER>:/tmp
scp <NAME_OF_PEM_DB_FILE>.keytab <OS_USERNAME_ON_PEM_SERVER>@<HOSTNAME_OF_PEM_SERVER>:/tmp
```

On the PEM server, move the key tables to the required location and change ownership:

```
mv /tmp/<NAME_OF_PEM_WEB_FILE>.keytab <PEM_INSTALLATION_DIRECTORY>/share
chown pem <PEM_INSTALLATION_DIRECTORY>/share/<NAME_OF_PEM_WEB_FILE>.keytab
```

```
mv /tmp/<NAME_OF_PEM_DB_FILE>.keytab <DATA_DIRECTORY_OF_POSTGRES>/
chown enterprisedb <DATA_DIRECTORY_OF_POSTGRES>/<NAME_OF_PEM_DB_FILE>.keytab
```

Where:

- `NAME_OF_PEM_WEB_FILE` is the name specified for the key table for the PEM web application.
- `NAME_OF_PEM_DB_FILE` is the name specified for the key table for the PEM backend database server.
- `OS_USERNAME_ON_PEM_SERVER` is the name of the operating system user on the PEM server.
- `DATA_DIRECTORY_OF_POSTGRES` is the path of the data directory of the installed Postgres database (PostgreSQL/EDB Postgres Advanced Server).

### 4. Configure the PEM backend database server

Add the key table location in the `postgresql.conf` file:

```
krb_server_keyfile='FILE:/<DATA_DIRECTORY_OF_POSTGRES>/<NAME_OF_PEM_DB_FILE>.keytab'
```

Where:

- `NAME_OF_PEM_DB_FILE` is the name specified for the key table for the PEM backend database server.
- `DATA_DIRECTORY_OF_POSTGRES` is the path of the data directory of the installed Postgres database (PostgreSQL/EDB Postgres Advanced Server).

Edit the `krb5.conf` file:

```
$ sudo vim /etc/krb5.conf
[libdefaults]
    default_realm = EDBPEM.ORG
    Forwardable = True

[domain_realm]
    .edbpem.org = EDBPEM.ORG
    edbpem.org = EDBPEM.ORG
```

```
[realms]
EDBPEM.ORG = {
    kdc = krb5server.edbpem.org
    admin_server = krb5server.edbpem.org
}
```

Restart the database server to reflect the changes:

```
systemctl restart <POSTGRES_SERVICE_NAME>
```

`POSTGRES_SERVICE_NAME` is the service name of the Postgres (PostgreSQL/EDB Postgres Advanced Server) database, for example, `postgresql-13` for PostgreSQL 13 database on RHEL or Rocky Linux platforms.

## 5. Obtain and view the initial ticket

The kinit utility obtains and caches Kerberos tickets. You typically use this utility to obtain the ticket-granting ticket, entering a password to decrypt the credential from the key distribution center (KDC). The ticket-granting ticket is then stored in your credential cache.

You can view the details of the ticket using the klist utility.

### Note

Install the Kerberos client on the PEM server and the client machine before using kinit and klist.

```
$ kinit <USERNAME@REALM>
$ klist
```

It displays the principal along with the Kerberos ticket.

### Note

The `USERNAME@REALM` specified here must be a database user having the pem\_admin role and CONNECT privilege on `pem` database.

## 6. Configure the PEM server

Run the PEM configure script on the PEM server to use Kerberos authentication:

```
```shell
$ sudo PEM_APP_HOST=<HOSTNAME_OF_PEM_SERVER> PEM_KRB_KTNAME=
<PEM_INSTALLATION_DIRECTORY/share/<NAME_OF_PEM_WEB_FILE>.keytab <PEM_INSTALLATION_DIRECTORY>/bin/configure-
pem-server.sh
```
```

Configure `PEM_DB_HOST` in the `config_setup.py` file. Check that the value of `PEM_AUTH_METHOD` is set to `'kerberos'`.

```
```shell
$ sudo vim <PEM_INSTALLATION_DIRECTORY>/share/web/config_setup.py
PEM_DB_HOST=`<HOSTNAME_OF_PEM_SERVER>`
```
```



Configure the host in the `.install-config` file:

```
```shell
$ sudo vim <PEM_INSTALLATION_DIRECTORY>/share/.install-config
HOST=`<HOSTNAME_OF_PEM_SERVER>`
```
```

If the PEM server uses Kerberos authentication:

- All the monitored servers default to use the same authentication. To override the default, in the `config_local.py` file, add the parameter `ALLOW_DATABASE_CONNECTION_WITHOUT_KERBEROS` and set it to `True`.
- All the authenticated user principals are appended with the realm (USERNAME@REALM) and passed as the database user name by default. To override the default, in the `config_local.py` file, add the parameter `PEM_USER_KRB_INCLUDE_REALM` and set it to `False`.
- Restart the Apache server

```
sudo systemctl restart <SERVICE_NAME>
```

- Edit the entries at the top of `pg_hba.conf` to use the gss authentication method, and reload the database server.

```
host    pem          +pem_user    <ip_of_pem_server>/32    gss
host    postgres     +pem_user    <ip_of_pem_server>/32    gss
```

```
systemctl reload <POSTGRES_SERVICE_NAME>
```

`POSTGRES_SERVICE_NAME` is the service name of the Postgres (PostgreSQL/EDB Postgres Advanced Server) database, for example, `postgresql-13` for PostgreSQL 13 database on RHEL or Rocky Linux platforms.

#### Note

If you're using PostgreSQL or EDB Postgres Advanced Server 12 or later, then you can specify connection type as `hostgssenc` to allow only gss-encrypted connection.

## 7. Browser settings

Configure the browser on the client machine to access the PEM web client to use the Spnego/Kerberos.

For Mozilla Firefox:

1. Open the low-level Firefox configuration page by loading the `about:config` page.
2. In the search box, enter `network.negotiate-auth.trusted-uris`.
3. Double-click the `network.negotiate-auth.trusted-uris` preference and enter the hostname or the domain of the web server that's protected by Kerberos HTTP SPNEGO. Separate multiple domains and hostnames with a comma.
4. In the search box, enter `network.negotiate-auth.delegation-uris`.
5. Double-click the `network.negotiate-auth.delegation-uris` preference and enter the hostname or the domain of the web server that's protected by Kerberos HTTP SPNEGO. Separate multiple domains and hostnames with a comma.
6. Select **OK**.

For Google Chrome on Linux or MacOS:

- Add the `--auth-server-whitelist` parameter to the `google-chrome` command. For example, to run Chrome from a Linux prompt, run the `google-chrome` command as follows:

```
google-chrome --auth-server-whitelist =
"hostname/domain"
```

- After configuring the PEM server, you can access the PEM web interface in your browser. Navigate to:

```
https://<ip_address_of_PEM_server>:8443/pem
```

#### Note

You might see the following error while connecting to your Postgres cluster:

```
psql -h hostname template1 psql: GSSAPI continuation error: Unspecified GSS failure. Minor code may
provide more information GSSAPI continuation error: Key version is not available
```

Add encryption types to the keytab using ktutil or by recreating the Postgres keytab with all crypto systems from AD.

## 6.3.2 Configuring the PEM server to use Windows Active Directory domain services for Kerberos authentication (SSPI)

The Windows Active Directory domain service works with hostnames and not with IP addresses. To use single sign-on in PEM Server using Active Directory domain services, configure the following machines with hostnames using the DNS:

- Windows server (domain controller)
- PEM server (PEM web server and PEM backend database server)
- Client machine

For example, if the realm on Windows Active Directory is `edbpem.internal`, then you can set the Windows server hostname to `Krb5server.edbpem.internal`, the PEM server hostname to `pem.edbpem.internal`, and the client's hostname to `pg12.edbpem.internal`.

### 1. Install Active Directory, the PEM server, and the PEM backend database server

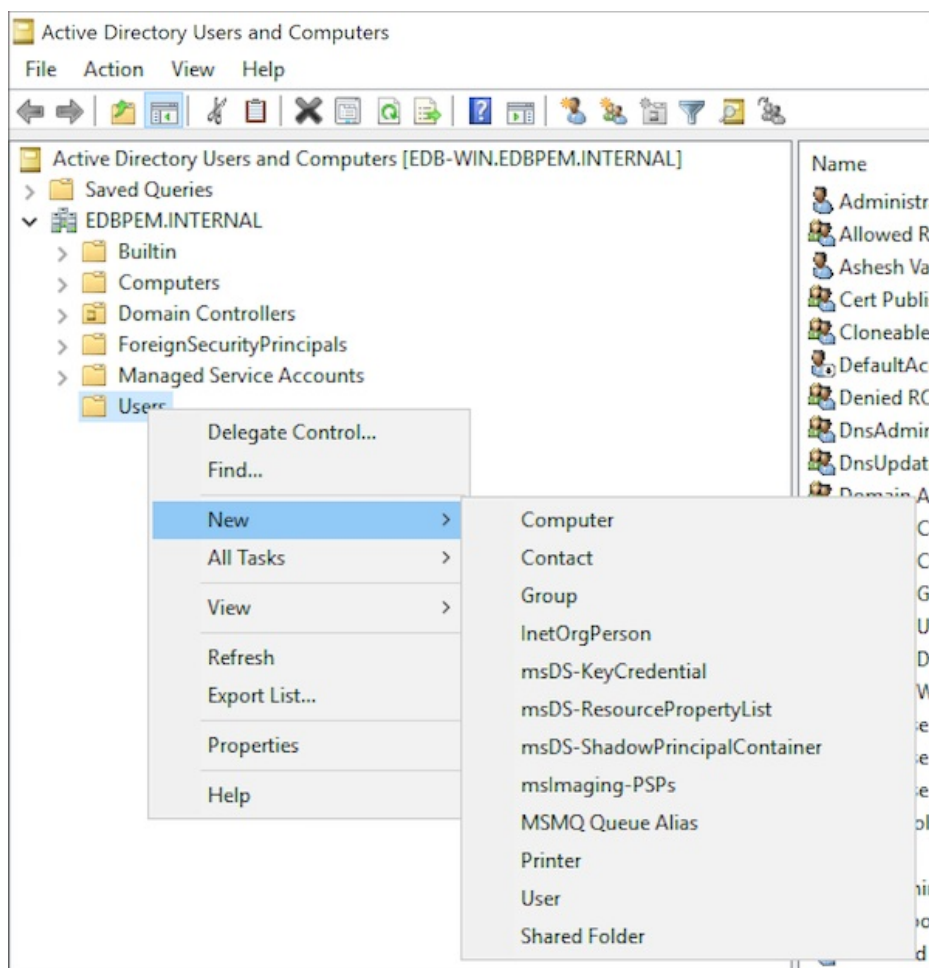
Perform the following installations:

- Install Active Directory on the Windows server (domain controller) that functions as the authentication server. Also, configure the Active Directory domain services to use Kerberos authentication, and then start it.
- Install the PEM server on a separate Linux machine. For more information, see [Installing the PEM server](#).
- Install the PEM backend database (Postgres/EDB Postgres Advanced Server) on the same Linux machine as the PEM server or a different one. For more information, see the installation steps on the [EDB Docs website](#).

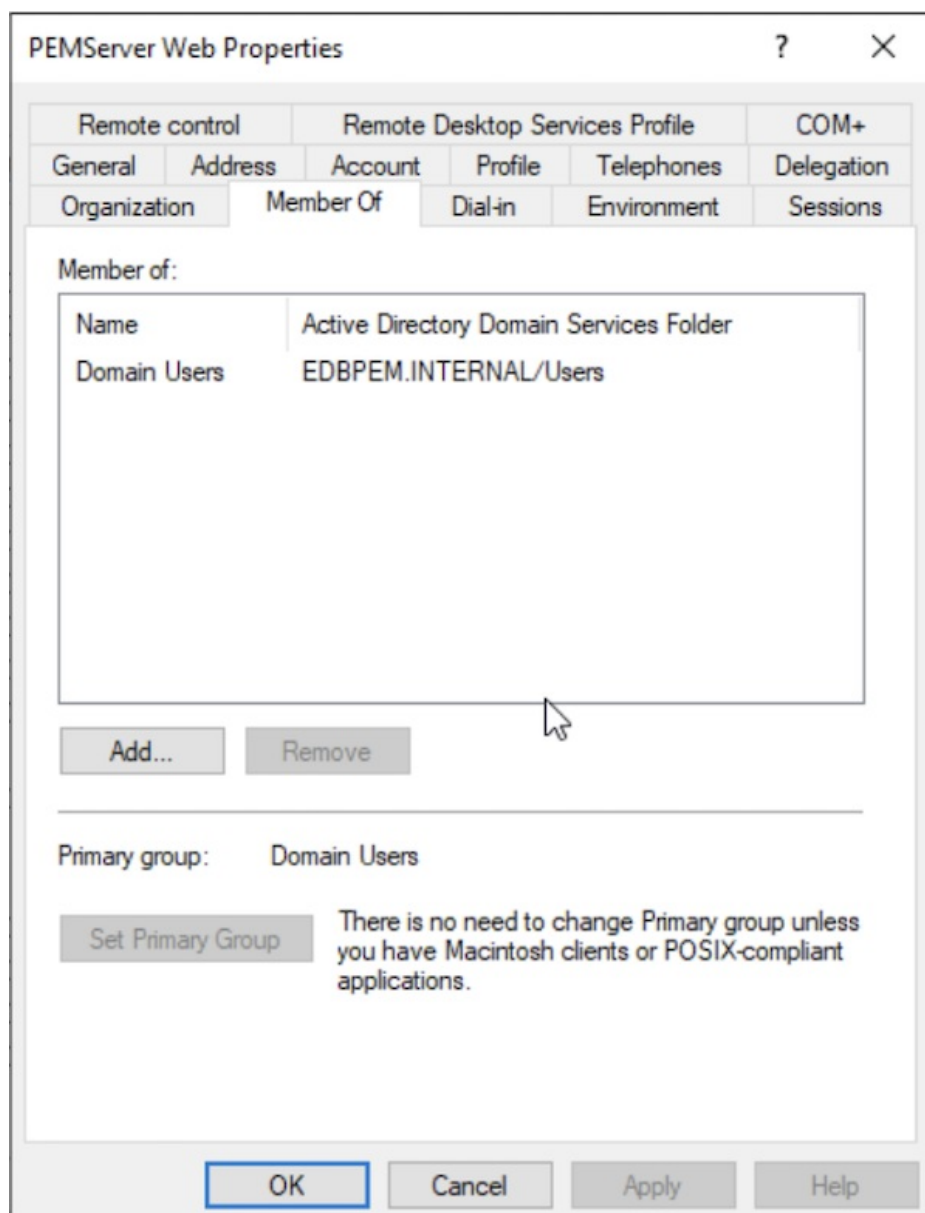
### 2. Create users in Active Directory to map with service principals

Create users in Active Directory of the Windows server to map with the HTTP service principal for the PEM web application.

1. Open **Active Directory Users and Computers** > **<DOMAIN\_NAME>** > **Users** Right-click and select **New** > **User**.



2. Enter the user details.
3. Give the password and make sure to clear **User must change password at next login**. Also select **User cannot change password** and **Password never expires**.
4. Review the user details.
5. On the PEMServer Web Properties dialog box, add the users as members of the Domain Users group:



6. Create the user (for example, pemserverdb) in Active Directory of the Windows server to map with the Postgres service principal for the PEM backend database.

### 3. Extract key tables from Active Directory

Extract the key tables for the service principals and map them with the respective domain users you created.

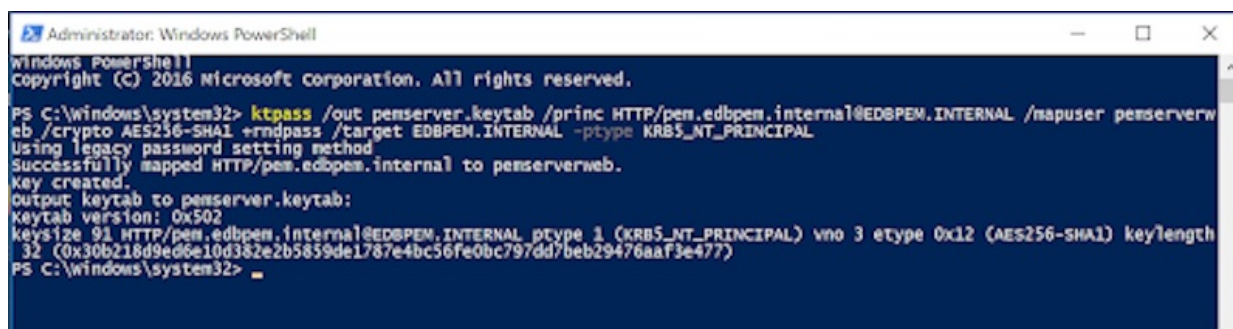
1. Open Windows PowerShell as an administrator. Create a key table for HTTP service principal mapping with the user pemserverweb and a key table for Postgres service principal mapping with the user pemserverdb:

```
ktpass /out pemserver.keytab /princ HTTP/pem.edbpem.internal@EDBPEM.INTERNAL /mapuser pemserverweb
/crypto AES256-SHA1 +rndpass /target EDBPEM.INTERNAL -ptype KRB5_NT_PRINCIPAL -kvno 0
ktpass /out pemdb.keytab /princ postgres/pem.edbpem.internal@EDBPEM.INTERNAL /mapuser pemserverdb
/crypto AES256-SHA1 +rndpass /target EDBPEM.INTERNAL -ptype KRB5_NT_PRINCIPAL -kvno 0
```

Where:

- `pemserver.keytab` is the name of the key table for the PEM web application.

- `pemdb.keytab` is the name of the key table for the PEM backend database server.
- `pem.edbpem.internal@EDBPEM.INTERNAL` is the hostname of the PEM server. Here, `@EDBPEM.INTERNAL` means `@REALM`.
- `pemserverweb` is the user for the PEM web application.
- `pemserverdb` is the user for the PEM backend database server.
- `EDBPEM.INTERNAL` is the domain of the target.



```

Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> ktpass /out pemserver.keytab /princ HTTP/pem.edbpem.internal@EDBPEM.INTERNAL /mapuser pemserverweb /crypto AES256-SHA1 +rndpass /target EDBPEM.INTERNAL -ptype KRB5_NT_PRINCIPAL
Using legacy password setting method
Successfully mapped HTTP/pem.edbpem.internal to pemserverweb.
Key created.
Output keytab to pemserver.keytab:
Keytab version: 0x502
keysize 91 HTTP/pem.edbpem.internal@EDBPEM.INTERNAL ptype 1 (KRB5_NT_PRINCIPAL) vno 3 etype 0x12 (AES256-SHA1) keylength 32 (0x30b218d9ed0e10d382e2b5859de1787e4bc56fe0bc797dd7beb29476aaf3e477)
PS C:\Windows\system32>

```

#### Note

The command line argument `+rndpass` resets the password for the domain user `pemserverweb` to a random password. The `/target` option is optional.

2. On the **Accounts** tab, add Kerberos support for the user accounts.

PEMServer Web Properties

|                |                                 |            |             |            |
|----------------|---------------------------------|------------|-------------|------------|
| Organization   | Member Of                       | Dial-in    | Environment | Sessions   |
| Remote control | Remote Desktop Services Profile |            |             | COM+       |
| General        | Address                         | Account    | Profile     | Telephones |
|                |                                 | Delegation |             |            |

User logon name:  
 @EDBPEM.INTERNAL

User logon name (pre-Windows 2000):

☐ Unlock account

Account options:

- ☐ Use only Kerberos DES encryption types for this account
- ☒ This account supports Kerberos AES 128 bit encryption.
- ☒ This account supports Kerberos AES 256 bit encryption.
- ☐ Do not require Kerberos preauthentication

Account expires:

☒ Never

☐ End of:

!!! Note On the **Accounts** tab, the user logon name shows `HTTP/pem.edbpem.internal@EDBPEM.INTERNAL`. The **Delegation** tab is enabled for the pemserverweb user.

- On the **Delegation** tab, select **Trust this user for delegation to any service (Kerberos only)** for the users you created.
- Copy both the key tables to the PEM server host or to the PEM web server and PEM backend database server hosts if installed on different hosts.
- On the PEM server, move the key tables to the required location and change the ownership:

```
mv /tmp/pemserver.keytab <PEM_INSTALLATION_DIRECTORY>/share
chown pem <PEM_INSTALLATION_DIRECTORY>/share/pemserver.keytab
```

```
mv /tmp/pemdb.keytab <DATA_DIRECTORY_OF_POSTGRES>/
chown enterprisedb <DATA_DIRECTORY_OF_POSTGRES>/pemdb.keytab
```

Where:

- `OS_USERNAME_ON_PEM_SERVER` is the name of the operating system user on the PEM server.
- `DATA_DIRECTORY_OF_POSTGRES` is the path of the data directory of the installed Postgres database (PostgreSQL/EDB Postgres Advanced Server).

## 4. Configure the PEM backend database server

Add the key table location in the `postgresql.conf` file.

```
krb_server_keyfile='FILE: /<DATA_DIRECTORY_OF_POSTGRES>/pemdb.keytab'
```

Where `DATA_DIRECTORY_OF_POSTGRES` is the path of the data directory of the installed Postgres database (PostgreSQL/EDB Postgres Advanced Server).

Edit the `krb5.conf` file:

```
``ini
$ sudo vim /etc/krb5.conf
[libdefaults]
default_realm = EDBPEM.INTERNAL
Forwardable = True

[domain_realm]
.edbpem.org = EDBPEM.INTERNAL
edbpem.org = EDBPEM.INTERNAL

[realms]
EDBPEM.INTERNAL = {
    kdc = krb5server.edbpem.internal
    admin_server = krb5server.edbpem.internal
}
...
```

Restart the database server to reflect the changes:

```
```shell
systemctl restart <POSTGRES_SERVICE_NAME>
```
```

`POSTGRES_SERVICE_NAME` is the service name of the Postgres (PostgreSQL/EDB Postgres Advanced Server) database, for example, `postgresql-13` for PostgreSQL 13 database on RHEL or Rocky Linux platforms.

## 5. Obtain and view the initial ticket

The `kinit` utility obtains and caches Kerberos tickets. You typically use this utility to obtain the ticket-granting ticket, using a password you entered to decrypt the credential from the key distribution center (KDC). The ticket-granting ticket is then stored in your credential cache.

You can view the details of the ticket using the `klist` utility.

### Note

You must install the Kerberos client on the PEM server and the client machine before using `kinit` and `klist`.

```
$ kinit <USERNAME@REALM>
$ klist
```

It displays the principal along with the Kerberos ticket.

**Note**

The `USERNAME@REALM` specified here must be a database user having the `pem_admin` role and `CONNECT` privileges on the `pem` database.

## 6. Configure the PEM server

Run the PEM configure script on the PEM server to use Kerberos authentication:

```
```shell
$ sudo PEM_APP_HOST=pem.edbpem.internal PEM_KRB_KTNAME=<PEM_INSTALLATION_DIRECTORY>/share/pemserver.keytab
<PEM_INSTALLATION_DIRECTORY>/bin/configure-pem-server.sh
```
```

In the `config_setup.py` file, configure `PEM_DB_HOST` and check that the value of `PEM_AUTH_METHOD` is set to `'kerberos'`.

```
```shell
$ sudo vim <PEM_INSTALLATION_DIRECTORY>/share/web/config_setup.py
PEM_DB_HOST=`pem.edbpem.internal`
```
```

Configure `HOST` in the `.install-config` file.

```
```shell
$ sudo vim <PEM_INSTALLATION_DIRECTORY>/share/.install-config
HOST=`pem.edbpem.internal`
```
```

If the PEM server uses Kerberos authentication:

- All the monitored servers default to use the same authentication. To override the default, in the `config_local.py` file, add the parameter `ALLOW_DATABASE_CONNECTION_WITHOUT_KERBEROS` and set it to `True`.
- All the authenticated user principals are appended with the realm (`USERNAME@REALM`) and passed as the database user name by default. To override the default, in the `config_local.py` file, add the parameter `PEM_USER_KRB_INCLUDE_REALM` and set it to `False`.

Restart the Apache server:

```
```shell
sudo systemctl restart <SERVICE_NAME>
```
```

Edit the entries at the top in `pg_hba.conf` to use the gss authentication method. Then reload the database server.

```
```shell
host pem      +pem_user    <ip_of_pem_server>/32    gss
host postgres +pem_user    <ip_of_pem_server>/32    gss
```

```shell
systemctl reload <POSTGRES_SERVICE_NAME>
```
```



`POSTGRES_SERVICE_NAME` is the service name of the Postgres (PostgreSQL/EDB Postgres Advanced Server) database, for example, `postgresql-13` for PostgreSQL 13 database on RHEL or Rocky Linux platforms.

#### Note

You can't specify the connection type as `hostgssenc`. Windows doesn't support gss encrypted connection.

## 7. Browser settings

Configure the browser on the client machine to access the PEM web client to use the Spnego/Kerberos.

For Mozilla Firefox:

- Open the low-level Firefox configuration page by loading the `about:config` page.
- In the search box, enter `network.negotiate-auth.trusted-uris`.
- Double-click the `network.negotiate-auth.trusted-uris` preference and enter the hostname or the domain of the web server that's protected by Kerberos HTTP SPNEGO. Separate multiple domains and hostnames with commas.
- In the search box, enter `network.negotiate-auth.delegation-uris`.
- Double-click the `network.negotiate-auth.delegation-uris` preference and enter the hostname or the domain of the web server that's protected by Kerberos HTTP SPNEGO. Separate multiple domains and hostnames with commas.
- Select OK.

For Google Chrome on Linux or MacOS:

- Add the `--auth-server-whitelist` parameter to the `google-chrome` command. For example, to run Chrome from a Linux prompt, run the `google-chrome` command as follows:

```
google-chrome --auth-server-whitelist =
"hostname/domain"
```

- After configuring the PEM server, you can access the PEM web interface in your browser. Navigate to:

```
https://<ip_address_of_PEM_server>:8443/pem
```

#### Note

You might see the following error while connecting to your Postgres cluster:

```
psql -h hostname template1 psql: GSSAPI continuation error: Unspecified GSS failure. Minor code may
provide more information GSSAPI continuation error: Key version is not available
```

Add encryption types to the keytab using `ktutil` or by recreating the Postgres keytab with all crypto systems from AD.

## 6.3.3 Configuring the PEM server to use two-factor authentication

#### New Feature

Two-factor authentication (2FA) support is available in PEM version 8.2.0 and later.

PEM supports two methods for 2FA:

- Email authentication
- Authenticator app (such as Google Authenticator)

To enable 2FA, you can copy these settings from the `config.py` file to the `config_local.py` file and modify the following parameters.

| Parameter              | Description   |
|------------------------|---|
| MFA_ENABLED            | Set to <code>true</code> to enable the two-factor authentication. Default value is <code>false</code> .   |
| MFA_FORCE_REGISTRATION | Set to <code>true</code> to ask the users to register forcefully for the two-factor authentication methods at login. Default value is <code>false</code> .                              |
| MFA_SUPPORTED_METHODS  | Set to <code>email</code> to use the email authentication method (send a one-time code by email) or <code>authenticator</code> to use the TOTP-based application authentication method. |
| MFA_EMAIL_SUBJECT      | Set to the subject of the email for email authentication. Default value is <code>&lt;APP_NAME&gt; - Verification Code</code> .  |

## Mail server configuration

To use the email authentication method, you need to configure mail server settings.

PEM server can send an email using either the SMTP configurations saved in the PEM configuration or using Flask-Mail.

To send the email verification code using the internal SMTP configuration from the PEM configuration, set the parameter `MAIL_USE_PEM_INTERNAL` to `True`. If set to `False`, the following mail configuration is used to send the code on the user-specified email address:

- `MAIL_SERVER = 'localhost'`
- `MAIL_PORT = 25`
- `MAIL_USE_TLS = False`
- `MAIL_USE_SSL = False`
- `MAIL_USERNAME = None`
- `MAIL_PASSWORD = None`
- `MAIL_DEFAULT_SENDER = None`

For more details about these configurations, see the [Flask-Mail documentation](#).

### Note

PEM SMTP alerts don't use this configuration.

## 6.4 Securing your deployment

When you configure PEM, there are security practices to consider. These security practices are optional. However, EDB recommends these practices to enhance the overall system security.

PEM depends on third-party components from the vendor repository, including the Apache web server, OpenSSL, snmp++, and libcurl. To ensure these components are up to date, update your operating system and regularly apply security updates to avoid any security vulnerability. Without the most recent security patches, your system is vulnerable to cyberattacks. Security patches protect your devices and their data by applying the latest updates to guard against the latest threats.

Some of the benefits of regularly applying security patches include:

- Reducing exposure to cyberattacks
- Avoiding lost productivity
- Protecting data from malware like ransomware

- Avoiding worm infections that use security loopholes to spread over the network

## 6.4.1 Apache HTTPD security configurations

On Windows, Apache HTTPD is named PEM HTTPD. The Apache HTTPD configuration file is `pme.conf` and the SSL configuration file is `httpd-ssl-pem.conf`. Both configuration files are in the `<Apache_Installation_Path>/conf/addons` directory.

On Linux, the Apache HTTPD configuration file is `edb-pem.conf` and the SSL configuration file is `edb-ssl-pem.conf`. Both configurations files are in the `<Apache_Installation_Path>/conf.d` directory.

### Disable SSLv2 and SSLv3

Disable SSL versions SSLv2, SSLv3, TLS 1, and TLS 1.1. These versions are the most vulnerable and have cryptographic concerns.

To disable the versions, add the following command to the Apache HTTPD configuration file:

```
SSLProtocol -ALL +TLSv1.2
```

Restart the web server to apply the changes to the configuration file.

PEM adds the following lines to the SSL configuration file to allow for the use of TLS 1.2 as security:

```
SSLProtocol -All TLSv1.2
```

```
SSLProxyProtocol -All TLSv1.2
```

### Secure HTTPD with SSL certificates

We recommend an additional layer of SSL security for the web application.

During PEM installation, PEM generates and uses self-signed certificates for the Apache/HTTPD server. If you want to use your own SSL certificate for PEM, you must update the Apache configuration file.

On Linux, you need to open the Apache HTTPD configuration file (`edb-ssl-pem.conf`) in a text editor and be a user with write permission on the file. You must also change the server name and file names in the configuration file to match your certificate files.

Update these two SSL directives in the PEM VirtualHost section:

- `SSLCertificateFile` is your DigiCert certificate file (for example, `your_domain_name.crt`).
- `SSLCertificateKeyFile` is the `.key` file generated when you created the CSR (for example, `your_private.key`).

For example, make the following updates:

```
SSLEngine on
SSLCertificateFile /path/to/your_domain_name.crt
SSLCertificateKeyFile /path/to/your_private.key
```

You can also replace the HTTPD self-signed SSL certificates with trusted CA-signed certificates in PEM. For instructions, see this [Postgres tutorial](#).

## Disable web server information exposure

We recommend that you disable all web server signatures as part of web server security. To disable the web server signature, add the following parameters to the Apache HTTPD configuration file. PEM disables exposure of the information by adding the parameters to the Apache HTTPD configuration file.

```
ServerTokens Prod
ServerSignature Off
```

The `ServerTokens` directive controls the server response header field, which returns to the client. We recommend that you hide the Apache server version by adding this parameter in the Apache HTTPD configuration file.

The `ServerSignature` directive includes a footer for server-produced documents. The footer contains information regarding the Apache configuration, like the Apache and operating system version. To limit the exposure of this information, we recommend that you disable the directive in the Apache HTTPD configuration file.

Restart the web server to apply any changes to the Apache HTTPD configuration file.

## Disable directory listing

The directory listing allows an attacker to view the complete contents of the directory. By default, the web server enables this listing, allowing an attacker to discover and view any file. This listing might lead to the attacker reverse engineering an application to obtain the source code, analyze it for possible security flaws, and discover more information about an application.

To avoid this risk, disable the directory listing by setting the `Options` directive in the Apache HTTPD configuration file. PEM disables the directory listing by setting the following option in the web server configuration file:

```
<Directory /application/directory> Options -Indexes </Directory>
```

Restart the web server to apply the changes made to the configuration file.

## Restrict the access to a network or IP address

Apache allows you to provide the client hostnames or IP addresses that can access the application. To give access to an IP address or network, enter the network address in the `Allow` directive.

```
<Directory /application/hostname>
Options None
AllowOverride None
Order deny,allow
Deny from all
Allow from 192.168.0.0/24
</Directory>
```

The `ALLOWED_HOSTS` configuration parameter in the application configuration file sets the allowed hosts by IP address. The application configuration file (`config_local.py`) is located in `<PEM_INSTALLATION_PATH>/web`.

PEM allows all the hosts to connect with the application.

For example:

```
# You can set the range of IP addresses in the configuration
file:
ALLOWED_HOSTS = ['225.0.0.0/8', '226.0.0.0/7',
'228.0.0.0/6']

#You can set the IP addresses to allow a host on a subnet level in the configuration
file:
ALLOWED_HOSTS = ['192.0.2.0/28',
'::192.0.2.0/124']

#You can set a specific individual host address (based on the IP address) in the configuration
file:
ALLOWED_HOSTS = ['127.0.0.1',
'192.168.0.1']
```

To apply the application configuration file changes, restart the web server.

## Cross-site tracing

You can use two HTTP methods to debug the web server connections: TRACE and TRACK. When an HTTP TRACE request is sent to a supported web server, the server responds and echoes the data passed to it, including any HTTP headers. We recommend that you disable these methods in the Apache configuration.

To disable the TRACE method for all virtual hosts, add the following line to the Apache HTTPD configuration file:

```
TraceEnable off
```

To disable these methods for a specific virtual host, add the following lines for each virtual host in the Apache configuration file. PEM also adds the following lines to the Apache HTTPD configuration file:

```
RewriteEngine on
RewriteCond %{REQUEST_METHOD} ^(TRACE|TRACK|OPTIONS)
RewriteRule .* - [F]
```

## Run web server from a nonprivileged user account

Running the Apache web service as a root user can create a security issue. We recommend that you run the web server as a unique nonprivileged user. Doing so helps to secure any other services running during a security breach.

PEM runs as a WSGI application. To delegate the WSGI applications that are running, create distinct daemon processes using the `WSGIDaemonProcess` directive.

On Linux, the Apache web server starts as the root user. Meanwhile, PEM runs the daemon processes as the pem user. On Windows, the `WSGIDaemonProcess` directive and features aren't available. During the installation, PEM HTTPD installs as a service. For this service installation to succeed, you must be a member of the Administrators group.

By default, the Apache services run as the system user (the `LocalSystem` account).

## Customize security HTTP headers in PEM web server

PEM contains a separate configuration file to fix the following security issues. We recommend that you override the configuration only of `config_local.py` and not of `config.py`. The `config_local.py` file is present only on some systems. You must create it to override the application-level configurations. During a PEM upgrade, you can't overwrite `config_local.py`, but you can overwrite changes in `config.py` and `config_distro.py`. Remove `config_local.py` after uninstalling PEM.

By default, `config_local.py` is in `/usr/edb/pem/web` on Linux and at `C:\ProgramFiles\edb\pem\server\share\web` on Windows.

## Host header injection attacks

HTTP host header attacks exploit vulnerable websites that handle the host header value in an unsafe way. If the server implicitly trusts the host header and fails to validate or escape it properly, an attacker can use this input to inject harmful payloads that manipulate server-side behavior. The web applications typically don't know the domain they are deployed on unless specified in a configuration file during setup. When they need to know the current domain, for example, they can resort to retrieving the domain from the host header to generate an absolute URL. The host header is a potential vector for exploiting a range of other vulnerabilities, most notably web cache poisoning and SQL injections.

### X-Frame-Options

X-Frame-Options indicate whether a browser is allowed to render a page in an `<iframe>` tag. It specifically protects against clickjacking. PEM has a host validation `X_FRAME_OPTIONS` option to prevent these attacks, which you can configure in the `config_local.py` file. The default is:

```
X_FRAME_OPTIONS = "SAMEORIGIN"
```

### Content-Security-Policy

Content-Security-Policy is part of the HTML5 standard. It provides a broader range of protection than the X-Frame-Options header, which it replaces. It is designed so that website authors can whitelist domains. The authors can load resources (like scripts, stylesheets, and fonts) from the whitelisted domains and also from domains that can embed a page.

PEM has a host validation `CONTENT_SECURITY_POLICY` option to prevent attacks, which you can configure in the `config_local.py` file. The default is:

```
CONTENT_SECURITY_POLICY = "default-src https: data: blob: 'unsafe-inline' 'unsafe-eval';"
```

### Strict-Transport-Security

The Strict-Transport-Security (HSTS) response header can prevent a man-in-the-middle attack. When you enable the option, websites or web applications tell browsers that they accept only HTTPS and not HTTP. The default is:

```
STRICT_TRANSPORT_SECURITY = "max-age=31536000;includeSubDomains"
```

#### Note

Adding this parameter can cause problems if config is changed. Therefore, we recommend that you add it only after PEM installation is complete and tested.

## X-Content-Type-Options

The X-Content-Type-Options response HTTP header is a marker. The server uses the marker to indicate that the MIME types advertised in Content-Type headers can't be changed and followed. The following is a way to opt out of MIME type sniffing, that is, to say that the MIME types are deliberately configured. The default is:

```
X_CONTENT_TYPE_OPTIONS = "nosniff"
```

## X-XSS-Protection

Cross-site scripting (XSS) is one of the most common application layer vulnerabilities in the web servers. XSS enables attackers to inject client-side scripts into web pages that other users view. The HTTP X-XSS-Protection response to the header is a feature of Internet Explorer, Chrome, and Safari. It stops pages from loading when they detect reflected cross-site scripting (XSS) attacks. These protections are unnecessary in modern browsers when sites implement a strong Content-Security-Policy that disables the use of inline JavaScript ('unsafe-inline'). However, these protections can still provide protections for users of older web browsers that don't yet support CSP. The default is:

```
X_XSS_PROTECTION = "1;  
mode=block"
```

To avoid this, add the following options to the Apache configuration file:

```
<IfModule mod_headers.c>  
Header set X-XSS-Protection "1; mode=block"  
</IfModule>
```

To apply the configuration file changes, restart the web server.

By default, PEM sets `X-XSS-Protection` to `"1; mode=block"` in the application configuration file, which is located at `/usr/edb/pem/web/config.py`.

To apply the changes, restart the Apache service.

For detailed information on the `config.py` file, see [Managing configuration settings](#).

## Cookie security

Cookies are small packets of data that a server sends to your browser to store configuration data. The browser sends them and all other requests to the same server, so it's important to know how to secure cookies. Multiple configuration options in `config.py` can make cookies secure. These are the three most important options:

- `SESSION_COOKIE_SECURE` — The flag prevents cookies from sending over an unencrypted connection. The browser can't add the cookie to any request to a server without an encrypted channel. The browser can add cookies only to connections such as HTTPS. The default is:

```
SESSION_COOKIE_SECURE = True
```

- `SESSION_COOKIE_HTTPONLY` — By default, JavaScript can read the content of cookies. The `HTTPOnly` flag prevents scripts from reading the cookie. Instead, the browser uses the cookie only with HTTP or HTTPS requests. Hackers can't exploit XSS vulnerabilities to learn the contents of the cookie. For example, the `sessionId` cookie never requires that it be read with a client-side script. So, you can set the `HTTPOnly` flag for `sessionId` cookies. The default is:

```
SESSION_COOKIE_HTTPONLY = True
```

- `ENHANCED_COOKIE_PROTECTION` — When you set this option to `True`, then a token is generated according to the IP address and user agent. In all subsequent requests, the token recalculates and compares to the one computed for the first request. If the session cookie is stolen and the

attacker uses it from another location, the generated token is different. In that case, the extension clears the session and blocks the request. The default is:

```
ENHANCED_COOKIE_PROTECTION = True
```

!!! Note This option can cause problems when the server deploys in dynamic IP address hosting environments, such as Kubernetes or behind load balancers. In such cases, set this option to `False`.

To apply the changes, restart the Apache service.

For detailed information on `config.py` file, see [Managing Configuration Settings](#).

## 6.4.2 PEM application security configurations

### Session timeout

Insufficient session expiration by the web application increases the exposure of other session-based attacks. The attacker has more time to reuse a valid session ID and hijack the associated session. The shorter the session interval is, the less time an attacker has to use the valid session ID. We recommend that you set the inactivity timeout for the web application to a low value to avoid this security issue.

In PEM, you can set the timeout value for a user session. When there's no user activity for a specified duration on the web console, PEM logs out the user from the web console. A PEM administrator can set the length of time for inactivity. This value is for the whole application and not for each user. To configure the timeout duration, modify the `USER_INACTIVITY_TIMEOUT` parameter in the `config_local.py` file, located in the `<PEM_INSTALLATION_PATH>/web` directory. By default, this functionality is disabled.

For example, to specify for an application to log out a user after 15 minutes of inactivity, set:

```
USER_INACTIVITY_TIMEOUT = 900
```

#### Note

The timeout value is specified in seconds.

To apply the changes, restart the Apache service.

For detailed information on the `config.py` file, see [Managing Configuration Settings](#).

### RestAPI header customization

You can customize the RestAPI token headers to meet your requirements. The default values aren't exposed by the `config.py` file. Customize the following headers in the `config_local.py` file:

#### PEM\_HEADER\_SUBJECT\_TOKEN\_KEY

This configuration option allows you to change the HTTP header name to get the generated token. By default, when you send a request to create a token, the server response has an `X-Subject-Token` header. This header contains the value of a newly generated token. If you want to customize the header name, then you can update the `config_local.py` file:



```
PEM_HEADER_SUBJECT_TOKEN_KEY = 'Pem-RestAPI-Generate-Token'
```

This command produces the following output:

```
curl -ik -X POST -d '{"username":"enterprisedb","password":"edb"}' -H "Content-Type: application/json"
https://localhost:8443/pem/api/token/
HTTP/1.1 201 CREATED
Date: Thu, 29 Oct 2020 11:03:48 GMT
Server: Apache
Content-Length: 326
Pem-RestAPI-Generate-Token: 997aef95-d46d-4d84-932a-a80146eaf84f
```

## PEM\_HEADER\_TOKEN\_KEY

This configuration option allows you to change the HTTP request header name. With this header name, you can send the token to the PEM server. By default, when you send a request to generate a token, the token header name is `X-Auth-Token`. If you want to customize the RestAPI request header name, then you can update the `config_local.py` file:

```
PEM_HEADER_TOKEN_KEY = 'Pem-Token'
```

This setting allows you to send the token:

```
$ curl -Lk -X GET -H "Pem-Token: gw5rزالoxydp91tttd1c97w24b5sv60cllc24sxy9"
https://localhost:8443/pem/api/v4/agent
```

## PEM\_TOKEN\_EXPIRY

This configuration option allows you to change the PEM RestAPI token expiry time after it's generated. By default, the token expiry time is set to 20 minutes (1200 seconds). If you want to change the token expiry time to 10 minutes, then you can update the `config_local.py` file:

```
PEM_TOKEN_EXPIRY = 600
```

To apply the changes, restart the Apache service.

## Role-based access control in PEM

Role-based access control (RBAC) restricts application access based on a user's role in an organization and is one of the primary methods for access control. The roles in RBAC refer to the levels of access that users have to the application. Users are allowed to access only the information needed to do their jobs. Roles in PEM are inheritable and additive, rather than subtractive. In other words, as a PEM admin you need to grant the lowest level role to the user and then grant the roles the user needs to perform their job. For example, to give access only to SQL profiler:

```
CREATE ROLE user_sql_profiler WITH LOGIN NOSUPERUSER NOCREATEDB NOCREATEROLE INHERIT NOREPLICATION
CONNECTION LIMIT -1 PASSWORD 'xxxxxx';

GRANT pem_user, pem_comp_sqlprofiler TO user_sql_profiler;
```

For detailed information on roles, see [PEM Roles](#).

## SQL/Protect plugin

Often, preventing an SQL injection attack is the responsibility of the application developer, while the database administrator has little or no control over the potential threat. The difficulty for database administrators is that the application must have access to the data to function properly.

SQL/Protect is a module that allows a database administrator to protect a database from SQL injection attacks. SQL/Protect examines incoming queries for typical SQL injection profiles in addition to the standard database security policies.

Attackers can perpetrate SQL injection attacks with several different techniques. A specific signature characterizes each technique. SQL/Protect examines queries for unauthorized relations, utility commands, SQL tautology, and unbounded DML statements. SQL/Protect gives the control back to the database administrator by alerting the administrator to potentially dangerous queries and then blocking those queries.

#### Note

This plugin works only on the EDB Postgres Advanced Server server, so this is useful only when your PEM database is hosted on the EDB Postgres Advanced Server server.

For detailed information about the SQL Profiler plugin, see [SQL Profiler](#).

## Password management

One security tip for PEM administrative users is to change your PEM login passwords to something new regularly. Changing your password:

- Prevents breaches of multiple accounts
- Prevents constant access
- Prevents the use of saved passwords on a physically unsecured system
- Limits access gained by keystroke loggers

## Run pemAgent jobs with a non-root user

In most cases, pemAgent is installed as a root user and runs as a daemon process with root privileges. By default, PEM disables running the scheduled jobs/task. PEM provides support for running scheduled jobs as a non-root user by changing the pemAgent configuration file.

To run scheduled jobs as a non-root user, modify the entry for the `batch_script_user` parameter in the `agent.cfg` file and specify the user to run the script. You can either specify a non-root user or root user identity. If you don't specify a user, or the specified user doesn't exist, then the script doesn't execute. Restart the agent after modifying the file. If a non-root user is running `pemagent`, then the value of `batch_script_user` is ignored, and the same non-root user used for running the `pemagent` executes the script.

To invoke a script on a Windows system, set the registry entry for `AllowBatchJobSteps` to `true` and restart the PEM agent. PEM registry entries are located in:

```
HKEY_LOCAL_MACHINE\Software\EnterpriseDB\PEM\agent
```

## Changing the pemAgent and PEM backend database server certificates

By default, when you install PEM, the installer generates and uses self-signed certificates for the pemAgent and PEM database server. PemAgent uses these certificates when connecting to the PEM database server. To use your own SSL certificate for the pemAgent and PEM database server, see [Managing certificates](#).

#### Note

PEM doesn't support placing the SSL CA certificates at a custom location. Don't change the location of `ca_certificate.crt` and

```
ca_key.key.
```

## 6.5 Web server installation options

During the PEM server installation, you can specify your hosting preferences for the Apache web server.

### PEM server and Apache web server on separate hosts

1. Install the PEM server on both the hosts. See [Installing the PEM server](#).
2. Configure the PEM server host by selecting the **Database** option on the first host.
3. Configure an Apache web server by selecting the **Web Services** option on the second host.

For more information about configuring a PEM server, see [Configuring the PEM server on Linux](#).

#### Note

For production environments, best practice is to have the PEM server and Apache web server on separate hosts.

### PEM server and Apache web server on the same host

1. Install the PEM server. See [Installing the PEM server](#).
2. Run the configuration script. Select the **Web Services and Database** option to install the PEM server and Apache web server on the same host. See [Configuring the PEM server on Linux platforms](#).

#### Note

For nonproduction environments, best practice is to have the PEM server and Apache web server on same hosts.

## 7 Installing Postgres Enterprise Manager server

Select a link to access the applicable installation instructions:

### Linux [x86-64 \(amd64\)](#)

#### Red Hat Enterprise Linux (RHEL) and derivatives

- [RHEL 9, RHEL 8](#)
- [Oracle Linux \(OL\) 9, Oracle Linux \(OL\) 8](#)
- [Rocky Linux 9, Rocky Linux 8](#)
- [AlmaLinux 9, AlmaLinux 8](#)

## SUSE Linux Enterprise (SLES)

- [SLES 15, SLES 12](#)

## Debian and derivatives

- [Ubuntu 22.04, Ubuntu 20.04](#)
- [Debian 11](#)

## Linux IBM Power (ppc64le)

## Red Hat Enterprise Linux (RHEL) and derivatives

- [RHEL 9, RHEL 8](#)

## SUSE Linux Enterprise (SLES)

- [SLES 15, SLES 12](#)

## Windows

- [Windows Server 2019](#)

## 7.1 Prerequisites to install the PEM server on Linux

You can install PEM on a single server, or you can install the web application server and the backend database on two separate servers. You must prepare your servers for PEM installation.

After completing the prerequisites, [install](#) and [configure](#) PEM. If you're using two servers, install and configure PEM on both servers.

To install a PEM server on Linux, perform this preliminary configuration:

1. Install a [supported Postgres instance](#) for PEM to use as a backend database. You can install this instance on the same server as you will use for the PEM web application or a separate server. You can also use an existing Postgres instance providing it is configured as detailed in steps 2 and 3 below.
2. Configure authentication on the Postgres backend database by updating the `pg_hba.conf` file.

You must make the following changes manually, prior to configuration. (Additional changes are made to this file during [configuration](#).)

- To create the relations required for PEM, the PEM configuration script connects to the Postgres backend database as a superuser of your choice using password authentication. This requires you to permit your chosen superuser to authenticate using a password. This user must be able to connect from any location in which you run the configuration script. In practice, this means the server on which the backend database

is located and the server on which the PEM web application is to be installed, if they're different.

- To allow the chosen superuser to connect using password authentication, add a line to `pg_hba.conf` that allows `host` connections using `md5` or `scram-sha-256` authentication, for example, `host all superusername 127.0.0.1/32 scram-sha-256`.

#### Note

If you're using EDB Postgres Advanced Server, see [Modifying the pg\\_hba.conf file](#).

If you're using PostgreSQL, see [Client Authentication](#).

3. Verify that the `sslutils` extension is installed on your Postgres server. If you're using PostgreSQL or EDB Postgres Extended Server on RHEL/AlmaLinux/Rocky Linux or SLES, you also need to install the `hstore contrib` module.

- If you're using EDB Postgres Advanced Server, you can install the `sslutils` extension as follows, where `<x>` is the EDB Postgres Advanced server version.

For RHEL/AlmaLinux/Rocky Linux 8/9:

```
dnf install edb-as<x>-server-sslutils
```

For Debian/Ubuntu:

```
apt install edb-as<x>-server-sslutils
```

For SLES:

```
zypper install edb-as<x>-server-sslutils
```

- If you're using PostgreSQL, you can install the `sslutils` and, if required, `hstore` modules as follows, where `<x>` is the PostgreSQL version.

For RHEL/AlmaLinux/Rocky Linux 8/9:

```
dnf install sslutils_<x> postgresql<x>-contrib
```

For Debian/Ubuntu:

```
apt install postgresql-<x>-sslutils
```

For SLES:

```
zypper install sslutils_<x> postgresql<x>-contrib
```

- If you're using EDB Postgres Extended Server, you can install the `sslutils` and, if required, `hstore` modules as follows, where `<x>` is the EDB Postgres Extended Server version.

For RHEL/AlmaLinux/Rocky Linux 8/9:

```
dnf install edb-postgreextended<x>-sslutils edb-postgreextended<x>-contrib
```

For Debian/Ubuntu:

```
apt install edb-postgresextended-sslutils-<x>
```

For SLES:

```
zypper install edb-postgresextended<x>-sslutils edb-postgresextended<x>-contrib
```

#### For Debian and Ubuntu users

Debian 10 and Ubuntu 20 changed the requirements for accepting certificates.

- If you want to install the PEM agent on a machine with an old version of sslutils, then you must upgrade sslutils to 1.3. Version 1.3 has a 4096-bit RSA key and sha256 signature algorithm support added to it.
- If you don't upgrade sslutils to 1.3, then PEM agent might fail to connect to the PEM backend database server, and it might log the error "ca md too weak."

4. If you're using a firewall, allow access to port 8443 on the server where the PEM web application will be located:

For RHEL/Rocky Linux/AlmaLinux/SLES:

```
firewall-cmd --permanent --zone=public --add-port=8443/tcp
firewall-cmd --reload
```

For Debian/Ubuntu:

```
iptables -t filter -A INPUT -p TCP --dport 8443 -j ACCEPT
```

5. Make sure the components Postgres Enterprise Manager depends on, such as python3, libboost, openssl (1.0.2k or later), snmp++, and libcurl, are up to date on all servers:

For RHEL/AlmaLinux/Rocky Linux 8:

```
dnf upgrade
```

For Debian/Ubuntu:

```
apt-get update
```

For SLES:

```
zypper update
```

## 7.2 Dependencies of the PEM Server and Agent on Linux

The PEM Server and Agent packages for Linux have dependencies on various system libraries. These dependencies are detailed below for reference.

### Note

A PEM Agent is always installed alongside PEM Server, so all dependencies must be present on hosts where PEM Server (either the database or the web application) is installed.

Typically, PEM is built against the latest version of each dependency available from the vendor repository for a given platform and architecture. In some cases, PEM requires a newer version of a library than is available in the vendor repository. In these cases a newer version of the package, prefixed with `edb-` is sourced from EDB's repositories.

#### Note

This information is provided for reference. Packages from vendor repositories are not supported or patched by EDB. Refer to your operating system documentation or support provider for details of these packages.

Because these dependencies are updated frequently, the tables below are valid only for the latest patch release of PEM.

## Python 3 and mod\_wsgi

Python 3 and mod\_wsgi (a Python module for Apache HTTPD) are required for PEM Server.

Platform	Architecture	Python/mod_wsgi package	Python version	Python path
RHEL 7	x86_64	edb-python39/edb-python39-mod-wsgi	3.9	/usr/libexec/edb-python39/bin/python3
	ppc64le	edb-python39/edb-python39-mod-wsgi	3.9	/usr/libexec/edb-python39/bin/python3
RHEL 8	x86_64	edb-python310/edb-python310-mod-wsgi	3.10	/usr/libexec/edb-python310/bin/python3
	ppc64le	edb-python310/edb-python310-mod-wsgi	3.10	/usr/libexec/edb-python310/bin/python3
	s390x	edb-python310/edb-python310-mod-wsgi	3.10	/usr/libexec/edb-python310/bin/python3
RHEL 9	x86_64	python39/python39-mod_wsgi	3.9	/usr/bin/python3
	ppc64le	python39/python39-mod_wsgi	3.9	/usr/bin/python3
	s390x	python39/python39-mod_wsgi	3.9	/usr/bin/python3
SLES 12	x86_64	edb-python39/edb-python39-mod-wsgi	3.9	/usr/libexec/edb-python39/bin/python3
	ppc64le	edb-python39/edb-python39-mod-wsgi	3.9	/usr/libexec/edb-python39/bin/python3
	s390x	edb-python39/edb-python39-mod-wsgi	3.9	/usr/libexec/edb-python39/bin/python3
SLES 15	x86_64	edb-python310/edb-python310-mod-wsgi	3.10	/usr/libexec/edb-python310/bin/python3
	ppc64le	edb-python310/edb-python310-mod-wsgi	3.10	/usr/libexec/edb-python310/bin/python3
	s390x	edb-python310/edb-python310-mod-wsgi	3.10	/usr/libexec/edb-python310/bin/python3
Ubuntu 20	amd64	edb-python310/edb-python310-mod-wsgi	3.10	/usr/libexec/edb-python310/bin/python3
Ubuntu 22	amd64	python310/libapache2-mod-wsgi-py3	3.10	/usr/bin/python3
Debian 10	amd64	edb-python310/edb-python310-mod-wsgi	3.10	/usr/libexec/edb-python310/bin/python3

Platform	Architecture	Python/mod_wsgi package	Python version	Python path
Debian 11	amd64	edb-python310/edb-python310-mod-wsgi	3.10	/usr/libexec/edb-python310/bin/python3
Debian 12	amd64/arm64	python311/libapache2-mod-wsgi-py3	3.11	/usr/bin/python3

## OpenSSL

The PEM Server and Agent require OpenSSL.

Platform	Architecture	package-version
RHEL 7	x86_64	openssl-1.0.2k
	ppc64le	openssl-1.0.2k
RHEL 8	x86_64	openssl-1.1.1k
	ppc64le	openssl-1.1.1k
	s390x	openssl-1.1.1k
RHEL 9	x86_64	openssl-3.0.7
	ppc64le	openssl-3.0.7
	s390x	openssl-3.0.7
SLES 12	x86_64	openssl-1.0.2p
	ppc64le	openssl-1.0.2p
	s390x	openssl-1.0.2p
SLES 15	x86_64	openssl-1.1.1l
	ppc64le	openssl-1.1.1l
	s390x	openssl-1.1.1l
Ubuntu 20	amd64	openssl-1.1.1f
Ubuntu 22	amd64	openssl-3.0.2
Debian 10	amd64	openssl-1.1.1n
Debian 11	amd64	openssl-1.1.1w
Debian 12	amd64/arm64	openssl-3.0.11

## Libcurl

The PEM Agent requires libcurl.

Platform	Architecture	package-version
RHEL 7	x86_64	libcurl-pem-8.4.0
	ppc64le	libcurl-pem-8.4.0
RHEL 8	x86_64	libcurl-pem-8.4.0
	ppc64le	curl-7.61.1



Platform	Architecture	package-version
RHEL 9	s390x	curl-7.61.1
	x86_64	curl-7.76.1
	ppc64le	curl-7.76.1
SLES 12	s390x	curl-7.76.1
	x86_64	curl-8.0.1
	ppc64le	curl-8.0.1
SLES 15	s390x	curl-8.0.1
	x86_64	curl-8.0.1
	ppc64le	curl-8.0.1
Ubuntu 20	amd64	libcurl4-7.68.0
Ubuntu 22	amd64	libcurl4-7.81.0
Debian 10	amd64	libcurl4-7.64.0
Debian 11	amd64	libcurl4-7.74.0
Debian 12	amd64/arm64	libcurl4-7.88.1

## SNMP++

The PEM Agent requires SNMP++.

Platform	Architecture	package-version
RHEL 7	x86_64	snmp++-3.4.2
	ppc64le	snmp++-3.4.2
RHEL 8	x86_64	snmp++-3.4.2
	ppc64le	edb-snmp++-3.4.10
RHEL 9	s390x	edb-snmp++-3.4.7
	x86_64	edb-snmp++-3.4.10
	ppc64le	edb-snmp++-3.4.10
SLES 12	s390x	edb-snmp++-3.4.10
	x86_64	edb-snmp++-3.4.10
	ppc64le	edb-snmp++-3.4.10
SLES 15	s390x	edb-snmp++-3.4.7
	x86_64	edb-snmp++-3.4.10
	ppc64le	edb-snmp++-3.4.10
Ubuntu 20	amd64	edb-snmp++-3.4.10
Ubuntu 22	amd64	edb-snmp++-3.4.10
Debian 10	amd64	edb-snmp++-3.4.10
Debian 11	amd64	edb-snmp++-3.4.10
Debian 12	amd64/arm64	edb-snmp++-3.4.10

## Boost libraries

The PEM Agent requires the Boost libraries.

Platform	Architecture	package-version
RHEL 7	x86_64	boost169-system-1.69.0
	ppc64le	None boost package
RHEL 8	x86_64	boost169-system-1.69.0
	ppc64le	boost-system-1.66.0
	s390x	boost-system-1.66.0
RHEL 9	x86_64	boost-system-1.75.0
	ppc64le	boost-system-1.75.0
	s390x	boost-system-1.75.0
SLES 12	x86_64	libboost_system1_54_0-1.54.0
	ppc64le	libboost_system1_54_0-1.54.0
	s390x	libboost_system1_54_0-1.54.0
SLES 15	x86_64	libboost_regex1_66_1-1.66.0
	ppc64le	libboost_regex1_66_1-1.66.0
	s390x	libboost_regex1_66_1-1.66.0
Ubuntu 20	amd64	libboost-system1.71.0-1.71.0
Ubuntu 22	amd64	libboost-system1.74.0-1.74.0
Debian 10	amd64	libboost-system1.67.0-1.67.0
Debian 11	amd64	libboost-system1.74.0-1.74.0
Debian 12	amd64/arm64	libboost-system1.74.0-1.74.0

## 7.3 Installing Postgres Enterprise Manager server on Linux x86 (amd64)

Operating system-specific install instructions are described in the corresponding documentation:

### Red Hat Enterprise Linux (RHEL) and derivatives

- [RHEL 9](#)
- [RHEL 8](#)
- [Oracle Linux \(OL\) 9](#)
- [Oracle Linux \(OL\) 8](#)
- [Rocky Linux 9](#)
- [Rocky Linux 8](#)
- [AlmaLinux 9](#)

- [AlmaLinux 8](#)

## SUSE Linux Enterprise (SLES)

- [SLES 15](#)
- [SLES 12](#)

## Debian and derivatives

- [Ubuntu 22.04](#)
- [Ubuntu 20.04](#)
- [Debian 11](#)

### 7.3.1 Installing Postgres Enterprise Manager server on RHEL 9 or OL 9 x86\_64

You can install PEM on a single server, or you can install the web application server and the backend database on two separate servers. You must prepare your servers for PEM installation.

After fulfilling the prerequisites and completing the installation procedure described in the following steps, you must [configure](#) PEM. If you're using two servers, install and configure PEM on both servers.

#### Prerequisites

Before you begin the installation process:

1. Install a [supported Postgres instance](#) for PEM to use as a backend database.

You can install this instance on the same server to be used for the PEM web application or on a separate server. You can also use an existing Postgres instance if it is configured as detailed in the next steps.

2. Configure authentication on the Postgres backend database by updating the `pg_hba.conf` file.

Make the following changes manually, prior to configuration. (Additional changes are necessary during [configuration](#).)

- To create the relations required for PEM, the PEM configuration script connects to the Postgres backend database as a superuser of your choice using password authentication. This requires you to permit your chosen superuser to authenticate using a password. This user must be able to connect from any location where you run the configuration script. In practice, this means the server where the backend database is located and the server where the PEM web application is to be installed, if they're different.
- To allow the chosen superuser to connect using password authentication, add a line to `pg_hba.conf` that allows `host` connections using `md5` or `scram-sha-256` authentication, such as `host all superusername 127.0.0.1/32 scram-sha-256`.

#### Note

If you're using EDB Postgres Advanced Server, see [Modifying the pg\\_hba.conf file](#).

If you're using PostgreSQL, see [Client Authentication](#).

3. Verify that the `sslutils` extension is installed on your Postgres server.

If you're using PostgreSQL or EDB Postgres Extended Server on RHEL/AlmaLinux/Rocky Linux or SLES, you also need to install the `hstore contrib` module.

- If you're using EDB Postgres Advanced Server, you can install the `sslutils` extension as follows, where `<x>` is the EDB Postgres Advanced server version.

```
sudo dnf install edb-as<x>-server-sslutils
```

- If you're using PostgreSQL, you can install the `sslutils` and, if required, `hstore` modules as follows, where `<x>` is the PostgreSQL version.

```
sudo dnf install sslutils_<x> postgresql<x>-contrib
```

- If you're using EDB Postgres Extended Server, you can install the `sslutils` and, if required, `hstore` modules as follows, where `<x>` is the EDB Postgres Extended Server version.

```
sudo dnf install edb-postgresextended<x>-sslutils edb-postgresextended<x>-contrib
```

4. If you're using a firewall, allow access to port 8443 on the server where the PEM web application will be located:

```
firewall-cmd --permanent --zone=public --add-port=8443/tcp
firewall-cmd --reload
```

5. Make sure the components Postgres Enterprise Manager depends on are up to date on all servers. You can do this by updating the whole system using your package manager as shown below. If you prefer to update individual packages, a full list of dependencies is provided in [Dependencies of the PEM Server and Agent on Linux](#).

```
sudo dnf upgrade
```

## Install the package

```
sudo dnf -y install edb-pem
```

## Initial configuration

```
# You can configure the PEM server using the following command:
sudo /usr/edb/pem/bin/configure-pem-server.sh
```

For more details, see [Configuring the PEM server on Linux](#).

### Note

- The operating system user `pem` is created while installing the PEM server. The PEM server web application is a WSGI application, which

runs under Apache HTTPD. The pem application data and the session is saved to this user's home directory.

## Supported locales

Currently, the Postgres Enterprise Manager server and web interface support a locale of `English(US) en_US` and use of a period (.) as a language separator character. Using an alternate locale or a separator character other than a period might cause errors.

## 7.3.2 Installing Postgres Enterprise Manager server on RHEL 8 or OL 8 x86\_64

You can install PEM on a single server, or you can install the web application server and the backend database on two separate servers. You must prepare your servers for PEM installation.

After fulfilling the prerequisites and completing the installation procedure described in the following steps, you must [configure](#) PEM. If you're using two servers, install and configure PEM on both servers.

### Prerequisites

Before you begin the installation process:

1. Install a [supported Postgres instance](#) for PEM to use as a backend database.

You can install this instance on the same server to be used for the PEM web application or on a separate server. You can also use an existing Postgres instance if it is configured as detailed in the next steps.

2. Configure authentication on the Postgres backend database by updating the `pg_hba.conf` file.

Make the following changes manually, prior to configuration. (Additional changes are necessary during [configuration](#).)

- To create the relations required for PEM, the PEM configuration script connects to the Postgres backend database as a superuser of your choice using password authentication. This requires you to permit your chosen superuser to authenticate using a password. This user must be able to connect from any location where you run the configuration script. In practice, this means the server where the backend database is located and the server where the PEM web application is to be installed, if they're different.
- To allow the chosen superuser to connect using password authentication, add a line to `pg_hba.conf` that allows `host` connections using `md5` or `scram-sha-256` authentication, such as `host all superusername 127.0.0.1/32 scram-sha-256`.

#### Note

If you're using EDB Postgres Advanced Server, see [Modifying the pg\\_hba.conf file](#).

If you're using PostgreSQL, see [Client Authentication](#).

3. Verify that the `sslutils` extension is installed on your Postgres server.

If you're using PostgreSQL or EDB Postgres Extended Server on RHEL/AlmaLinux/Rocky Linux or SLES, you also need to install the `hstore contrib` module.

- If you're using EDB Postgres Advanced Server, you can install the `sslutils` extension as follows, where `<x>` is the EDB Postgres Advanced server version.

```
sudo dnf install edb-as<x>-server-sslutils
```

- If you're using PostgreSQL, you can install the `sslutils` and, if required, `hstore` modules as follows, where `<x>` is the PostgreSQL version.

```
sudo dnf install sslutils_<x> postgresql<x>-contrib
```

- If you're using EDB Postgres Extended Server, you can install the `sslutils` and, if required, `hstore` modules as follows, where `<x>` is the EDB Postgres Extended Server version.

```
sudo dnf install edb-postgresextended<x>-sslutils edb-postgresextended<x>-contrib
```

4. If you're using a firewall, allow access to port 8443 on the server where the PEM web application will be located:

```
firewall-cmd --permanent --zone=public --add-port=8443/tcp
```

```
firewall-cmd --reload
```

5. Make sure the components Postgres Enterprise Manager depends on are up to date on all servers. You can do this by updating the whole system using your package manager as shown below. If you prefer to update individual packages, a full list of dependencies is provided in [Dependencies of the PEM Server and Agent on Linux](#).

```
sudo dnf upgrade
```

## Install the package

```
sudo dnf -y install edb-pem
```

## Initial configuration

```
# You can configure the PEM server using the following command:
sudo /usr/edb/pem/bin/configure-pem-server.sh
```

For more details, see [Configuring the PEM server on Linux](#).

### Note

- The operating system user `pem` is created while installing the PEM server. The PEM server web application is a WSGI application, which runs under Apache HTTPD. The pem application data and the session is saved to this user's home directory.

## Supported locales

Currently, the Postgres Enterprise Manager server and web interface support a locale of `English(US) en_US` and use of a period (.) as a language separator character. Using an alternate locale or a separator character other than a period might cause errors.

### 7.3.3 Installing Postgres Enterprise Manager server on AlmaLinux 9 or Rocky

You can install PEM on a single server, or you can install the web application server and the backend database on two separate servers. You must prepare your servers for PEM installation.

After fulfilling the prerequisites and completing the installation procedure described in the following steps, you must [configure](#) PEM. If you're using two servers, install and configure PEM on both servers.

#### Prerequisites

Before you begin the installation process:

1. Install a [supported Postgres instance](#) for PEM to use as a backend database.

You can install this instance on the same server to be used for the PEM web application or on a separate server. You can also use an existing Postgres instance if it is configured as detailed in the next steps.

2. Configure authentication on the Postgres backend database by updating the `pg_hba.conf` file.

Make the following changes manually, prior to configuration. (Additional changes are necessary during [configuration](#).)

- To create the relations required for PEM, the PEM configuration script connects to the Postgres backend database as a superuser of your choice using password authentication. This requires you to permit your chosen superuser to authenticate using a password. This user must be able to connect from any location where you run the configuration script. In practice, this means the server where the backend database is located and the server where the PEM web application is to be installed, if they're different.
- To allow the chosen superuser to connect using password authentication, add a line to `pg_hba.conf` that allows `host` connections using `md5` or `scram-sha-256` authentication, such as `host all superusername 127.0.0.1/32 scram-sha-256`.

#### Note

If you're using EDB Postgres Advanced Server, see [Modifying the pg\\_hba.conf file](#).

If you're using PostgreSQL, see [Client Authentication](#).

3. Verify that the `sslutils` extension is installed on your Postgres server.

If you're using PostgreSQL or EDB Postgres Extended Server on RHEL/AlmaLinux/Rocky Linux or SLES, you also need to install the `hstore contrib` module.

- If you're using EDB Postgres Advanced Server, you can install the `sslutils` extension as follows, where `<x>` is the EDB Postgres Advanced server version.

```
sudo dnf install edb-as<x>-server-sslutils
```

- If you're using PostgreSQL, you can install the `sslutils` and, if required, `hstore` modules as follows, where `<x>` is the PostgreSQL version.

```
sudo dnf install sslutils_<x> postgresql<x>-contrib
```

- If you're using EDB Postgres Extended Server, you can install the `sslutils` and, if required, `hstore` modules as follows, where `<x>` is the EDB Postgres Extended Server version.

```
sudo dnf install edb-postgresexteneded<x>-sslutils edb-postgresexteneded<x>-contrib
```

4. If you're using a firewall, allow access to port 8443 on the server where the PEM web application will be located:

```
firewall-cmd --permanent --zone=public --add-port=8443/tcp
firewall-cmd --reload
```

5. Make sure the components Postgres Enterprise Manager depends on are up to date on all servers. You can do this by updating the whole system using your package manager as shown below. If you prefer to update individual packages, a full list of dependencies is provided in [Dependencies of the PEM Server and Agent on Linux](#).

```
sudo dnf upgrade
```

## Install the package

```
sudo dnf -y install edb-pem
```

## Initial configuration

# You can configure the PEM server using the following command:

```
sudo /usr/edb/pem/bin/configure-pem-server.sh
```

For more details, see [Configuring the PEM server on Linux](#).

### Note

- The operating system user pem is created while installing the PEM server. The PEM server web application is a WSGI application, which runs under Apache HTTPD. The pem application data and the session is saved to this user's home directory.

## Supported locales

Currently, the Postgres Enterprise Manager server and web interface support a locale of `English(US) en_US` and use of a period (.) as a language separator character. Using an alternate locale or a separator character other than a period might cause errors.

## 7.3.4 Installing Postgres Enterprise Manager server on AlmaLinux 8 or Rocky

You can install PEM on a single server, or you can install the web application server and the backend database on two separate servers. You must prepare your servers for PEM installation.

After fulfilling the prerequisites and completing the installation procedure described in the following steps, you must [configure](#) PEM. If you're using two servers, install and configure PEM on both servers.



## Prerequisites

Before you begin the installation process:

1. Install a [supported Postgres instance](#) for PEM to use as a backend database.

You can install this instance on the same server to be used for the PEM web application or on a separate server. You can also use an existing Postgres instance if it is configured as detailed in the next steps.

2. Configure authentication on the Postgres backend database by updating the `pg_hba.conf` file.

Make the following changes manually, prior to configuration. (Additional changes are necessary during [configuration](#).)

- To create the relations required for PEM, the PEM configuration script connects to the Postgres backend database as a superuser of your choice using password authentication. This requires you to permit your chosen superuser to authenticate using a password. This user must be able to connect from any location where you run the configuration script. In practice, this means the server where the backend database is located and the server where the PEM web application is to be installed, if they're different.
- To allow the chosen superuser to connect using password authentication, add a line to `pg_hba.conf` that allows `host` connections using `md5` or `scram-sha-256` authentication, such as `host all superusername 127.0.0.1/32 scram-sha-256`.

### Note

If you're using EDB Postgres Advanced Server, see [Modifying the pg\\_hba.conf file](#).

If you're using PostgreSQL, see [Client Authentication](#).

3. Verify that the `sslutils` extension is installed on your Postgres server.

If you're using PostgreSQL or EDB Postgres Extended Server on RHEL/AlmaLinux/Rocky Linux or SLES, you also need to install the `hstore contrib` module.

- If you're using EDB Postgres Advanced Server, you can install the `sslutils` extension as follows, where `<x>` is the EDB Postgres Advanced server version.

```
sudo dnf install edb-as<x>-server-sslutils
```

- If you're using PostgreSQL, you can install the `sslutils` and, if required, `hstore` modules as follows, where `<x>` is the PostgreSQL version.

```
sudo dnf install sslutils_<x> postgresql<x>-contrib
```

- If you're using EDB Postgres Extended Server, you can install the `sslutils` and, if required, `hstore` modules as follows, where `<x>` is the EDB Postgres Extended Server version.

```
sudo dnf install edb-postgresextended<x>-sslutils edb-postgresextended<x>-contrib
```

4. If you're using a firewall, allow access to port 8443 on the server where the PEM web application will be located:

```
firewall-cmd --permanent --zone=public --add-port=8443/tcp
```

```
firewall-cmd --reload
```

5. Make sure the components Postgres Enterprise Manager depends on are up to date on all servers. You can do this by updating the whole system

using your package manager as shown below. If you prefer to update individual packages, a full list of dependencies is provided in [Dependencies of the PEM Server and Agent on Linux](#).

```
sudo dnf upgrade
```

## Install the package

```
sudo dnf -y install edb-pem
```

## Initial configuration

```
# You can configure the PEM server using the following command:  
sudo /usr/edb/pem/bin/configure-pem-server.sh
```

For more details, see [Configuring the PEM server on Linux](#).

### Note

- The operating system user pem is created while installing the PEM server. The PEM server web application is a WSGI application, which runs under Apache HTTPD. The pem application data and the session is saved to this user's home directory.

## Supported locales

Currently, the Postgres Enterprise Manager server and web interface support a locale of `English(US) en_US` and use of a period (.) as a language separator character. Using an alternate locale or a separator character other than a period might cause errors.

## 7.3.5 Installing Postgres Enterprise Manager server on SLES 15 x86\_64

### Note

Postgres Enterprise Manager 8.3 and later is supported on SLES.

## Prerequisites

Before you begin the installation process:

1. Install a [supported Postgres instance](#) for PEM to use as a backend database.

You can install this instance on the same server to be used for the PEM web application or on a separate server. You can also use an existing Postgres instance if it is configured as detailed in the next steps.

2. Configure authentication on the Postgres backend database by updating the `pg_hba.conf` file.

Make the following changes manually, prior to configuration. (Additional changes are necessary during [configuration](#).)

- To create the relations required for PEM, the PEM configuration script connects to the Postgres backend database as a superuser of your choice using password authentication. This requires you to permit your chosen superuser to authenticate using a password. This user must be able to connect from any location where you run the configuration script. In practice, this means the server where the backend database is located and the server where the PEM web application is to be installed, if they're different.
- To allow the chosen superuser to connect using password authentication, add a line to `pg_hba.conf` that allows `host` connections using `md5` or `scram-sha-256` authentication, such as `host all superusername 127.0.0.1/32 scram-sha-256`.

#### Note

If you're using EDB Postgres Advanced Server, see [Modifying the pg\\_hba.conf file](#).

If you're using PostgreSQL, see [Client Authentication](#).

3. Verify that the `sslutils` extension is installed on your Postgres server.

If you're using PostgreSQL or EDB Postgres Extended Server on RHEL/AlmaLinux/Rocky Linux or SLES, you also need to install the `hstore contrib` module.

- If you're using EDB Postgres Advanced Server, you can install the `sslutils` extension as follows, where `<x>` is the EDB Postgres Advanced server version.

```
sudo zypper install edb-as<x>-server-sslutils
```

- If you're using PostgreSQL, you can install the `sslutils` and, if required, `hstore` modules as follows, where `<x>` is the PostgreSQL version.

```
sudo zypper install sslutils_<x> postgresql<x>-contrib
```

- If you're using EDB Postgres Extended Server, you can install the `sslutils` and, if required, `hstore` modules as follows, where `<x>` is the EDB Postgres Extended Server version.

```
sudo zypper install edb-postgresextended<x>-sslutils edb-postgresextended<x>-contrib
```

4. If you're using a firewall, allow access to port 8443 on the server where the PEM web application will be located:

```
firewall-cmd --permanent --zone=public --add-port=8443/tcp
```

```
firewall-cmd --reload
```

5. Make sure the components Postgres Enterprise Manager depends on are up to date on all servers. You can do this by updating the whole system using your package manager as shown below. If you prefer to update individual packages, a full list of dependencies is provided in [Dependencies of the PEM Server and Agent on Linux](#).

```
sudo zypper update
```

## Install the package

```
sudo zypper -n install edb-pem
```

## Initial configuration

```
# You can configure the PEM server using the following command:
sudo /usr/edb/pem/bin/configure-pem-server.sh
```

For more details, see [Configuring the PEM server on Linux](#).

### Note

- The operating system user pem is created while installing the PEM server. The PEM server web application is a WSGI application, which runs under Apache HTTPD. The pem application data and the session is saved to this user's home directory.

## Supported locales

Currently, the Postgres Enterprise Manager server and web interface support a locale of `English(US) en_US` and use of a period (.) as a language separator character. Using an alternate locale or a separator character other than a period might cause errors.

## 7.3.6 Installing Postgres Enterprise Manager server on SLES 12 x86\_64

### Note

Postgres Enterprise Manager 8.3 and later is supported on SLES.

## Prerequisites

Before you begin the installation process:

1. Install a [supported Postgres instance](#) for PEM to use as a backend database.

You can install this instance on the same server to be used for the PEM web application or on a separate server. You can also use an existing Postgres instance if it is configured as detailed in the next steps.

2. Configure authentication on the Postgres backend database by updating the `pg_hba.conf` file.

Make the following changes manually, prior to configuration. (Additional changes are necessary during [configuration](#).)

- To create the relations required for PEM, the PEM configuration script connects to the Postgres backend database as a superuser of your choice using password authentication. This requires you to permit your chosen superuser to authenticate using a password. This user must be able to connect from any location where you run the configuration script. In practice, this means the server where the backend database is located and the server where the PEM web application is to be installed, if they're different.
- To allow the chosen superuser to connect using password authentication, add a line to `pg_hba.conf` that allows `host` connections using `md5` or `scram-sha-256` authentication, such as `host all superusername 127.0.0.1/32 scram-sha-256`.

### Note

If you're using EDB Postgres Advanced Server, see [Modifying the pg\\_hba.conf file](#).

If you're using PostgreSQL, see [Client Authentication](#).

3. Verify that the `sslutils` extension is installed on your Postgres server.

If you're using PostgreSQL or EDB Postgres Extended Server on RHEL/AlmaLinux/Rocky Linux or SLES, you also need to install the `hstore contrib` module.

- If you're using EDB Postgres Advanced Server, you can install the `sslutils` extension as follows, where `<x>` is the EDB Postgres Advanced server version.

```
sudo zypper install edb-as<x>-server-sslutils
```

- If you're using PostgreSQL, you can install the `sslutils` and, if required, `hstore` modules as follows, where `<x>` is the PostgreSQL version.

```
sudo zypper install sslutils_<x> postgresql<x>-contrib
```

- If you're using EDB Postgres Extended Server, you can install the `sslutils` and, if required, `hstore` modules as follows, where `<x>` is the EDB Postgres Extended Server version.

```
sudo zypper install edb-postgresextended<x>-sslutils edb-postgresextended<x>-contrib
```

4. If you're using a firewall, allow access to port 8443 on the server where the PEM web application will be located:

```
firewall-cmd --permanent --zone=public --add-port=8443/tcp
firewall-cmd --reload
```

5. Make sure the components Postgres Enterprise Manager depends on are up to date on all servers. You can do this by updating the whole system using your package manager as shown below. If you prefer to update individual packages, a full list of dependencies is provided in [Dependencies of the PEM Server and Agent on Linux](#).

```
sudo zypper update
```

## Install the package

```
sudo zypper -n install edb-pem
```

## Initial configuration

```
# You can configure the PEM server using the following command:
sudo /usr/edb/pem/bin/configure-pem-server.sh
```

For more details, see [Configuring the PEM server on Linux](#).

### Note

- The operating system user `pem` is created while installing the PEM server. The PEM server web application is a WSGI application, which

runs under Apache HTTPD. The pem application data and the session is saved to this user's home directory.

## Supported locales

Currently, the Postgres Enterprise Manager server and web interface support a locale of `English(US) en_US` and use of a period (.) as a language separator character. Using an alternate locale or a separator character other than a period might cause errors.

## 7.3.7 Installing Postgres Enterprise Manager server on Ubuntu 22.04 x86\_64

You can install PEM on a single server, or you can install the web application server and the backend database on two separate servers. You must prepare your servers for PEM installation.

After fulfilling the prerequisites and completing the installation procedure described in the following steps, you must [configure](#) PEM. If you're using two servers, install and configure PEM on both servers.

### Prerequisites

Before you begin the installation process:

1. Install a [supported Postgres instance](#) for PEM to use as a backend database.

You can install this instance on the same server to be used for the PEM web application or on a separate server. You can also use an existing Postgres instance if it is configured as detailed in the next steps.

2. Configure authentication on the Postgres backend database by updating the `pg_hba.conf` file.

Make the following changes manually, prior to configuration. (Additional changes are necessary during [configuration](#).)

- To create the relations required for PEM, the PEM configuration script connects to the Postgres backend database as a superuser of your choice using password authentication. This requires you to permit your chosen superuser to authenticate using a password. This user must be able to connect from any location where you run the configuration script. In practice, this means the server where the backend database is located and the server where the PEM web application is to be installed, if they're different.
- To allow the chosen superuser to connect using password authentication, add a line to `pg_hba.conf` that allows `host` connections using `md5` or `scram-sha-256` authentication, such as `host all superusername 127.0.0.1/32 scram-sha-256`.

#### Note

If you're using EDB Postgres Advanced Server, see [Modifying the pg\\_hba.conf file](#).

If you're using PostgreSQL, see [Client Authentication](#).

3. Verify that the `sslutils` extension is installed on your Postgres server.

If you're using PostgreSQL or EDB Postgres Extended Server on RHEL/AlmaLinux/Rocky Linux or SLES, you also need to install the `hstore contrib` module.

- If you're using EDB Postgres Advanced Server, you can install the `sslutils` extension as follows, where `<x>` is the EDB Postgres Advanced server version.

```
sudo apt-get install edb-as<x>-server-sslutils
```

- If you're using PostgreSQL, you can install the `sslutils` and, if required, `hstore` modules as follows, where `<x>` is the PostgreSQL version.

```
sudo apt-get install postgresql-<x>-sslutils
```

- If you're using EDB Postgres Extended Server, you can install the `sslutils` and, if required, `hstore` modules as follows, where `<x>` is the EDB Postgres Extended Server version.

```
sudo apt-get install edb-postgresextended-sslutils-<x>
```

#### Note

Ubuntu 20 changed the requirements for accepting certificates.

- If you want to install the PEM agent on a machine with an old version of `sslutils`, then you must upgrade `sslutils` to 1.3. Version 1.3 has a 4096-bit RSA key and sha256 signature algorithm support added to it.
- If you don't upgrade `sslutils` to 1.3, then PEM agent might fail to connect to the PEM backend database server, and it might log the error "ca md too weak."

4. If you're using a firewall, allow access to port 8443 on the server where the PEM web application will be located:

```
iptables -t filter -A INPUT -p TCP --dport 8443 -j ACCEPT
```

5. Make sure the components Postgres Enterprise Manager depends on are up to date on all servers. You can do this by updating the whole system using your package manager as shown below. If you prefer to update individual packages, a full list of dependencies is provided in [Dependencies of the PEM Server and Agent on Linux](#).

```
sudo apt-get update
```

## Install the package

```
sudo apt-get -y install edb-pem
```

## Initial configuration

# You can configure the PEM server using the following command:  

```
sudo /usr/edb/pem/bin/configure-pem-server.sh
```

For more details, see [Configuring the PEM server on Linux](#).

#### Note

- The operating system user `pem` is created while installing the PEM server. The PEM server web application is a WSGI application, which runs under Apache HTTPD. The `pem` application data and the session is saved to this user's home directory.

## Supported locales

Currently, the Postgres Enterprise Manager server and web interface support a locale of `English(US) en_US` and use of a period (.) as a language separator character. Using an alternate locale or a separator character other than a period might cause errors.

## 7.3.8 Installing Postgres Enterprise Manager server on Ubuntu 20.04 x86\_64

You can install PEM on a single server, or you can install the web application server and the backend database on two separate servers. You must prepare your servers for PEM installation.

After fulfilling the prerequisites and completing the installation procedure described in the following steps, you must [configure](#) PEM. If you're using two servers, install and configure PEM on both servers.

### Prerequisites

Before you begin the installation process:

1. Install a [supported Postgres instance](#) for PEM to use as a backend database.

You can install this instance on the same server to be used for the PEM web application or on a separate server. You can also use an existing Postgres instance if it is configured as detailed in the next steps.

2. Configure authentication on the Postgres backend database by updating the `pg_hba.conf` file.

Make the following changes manually, prior to configuration. (Additional changes are necessary during [configuration](#).)

- To create the relations required for PEM, the PEM configuration script connects to the Postgres backend database as a superuser of your choice using password authentication. This requires you to permit your chosen superuser to authenticate using a password. This user must be able to connect from any location where you run the configuration script. In practice, this means the server where the backend database is located and the server where the PEM web application is to be installed, if they're different.
- To allow the chosen superuser to connect using password authentication, add a line to `pg_hba.conf` that allows `host` connections using `md5` or `scram-sha-256` authentication, such as `host all superusername 127.0.0.1/32 scram-sha-256`.

#### Note

If you're using EDB Postgres Advanced Server, see [Modifying the pg\\_hba.conf file](#).

If you're using PostgreSQL, see [Client Authentication](#).

3. Verify that the `sslutils` extension is installed on your Postgres server.

If you're using PostgreSQL or EDB Postgres Extended Server on RHEL/AlmaLinux/Rocky Linux or SLES, you also need to install the `hstore contrib` module.

- If you're using EDB Postgres Advanced Server, you can install the `sslutils` extension as follows, where `<x>` is the EDB Postgres Advanced server version.

```
sudo apt-get install edb-as<x>-server-sslutils
```



- If you're using PostgreSQL, you can install the `sslutils` and, if required, `hstore` modules as follows, where `<x>` is the PostgreSQL version.

```
sudo apt-get install postgresql-<x>-sslutils
```

- If you're using EDB Postgres Extended Server, you can install the `sslutils` and, if required, `hstore` modules as follows, where `<x>` is the EDB Postgres Extended Server version.

```
sudo apt-get install edb-postgresextended-sslutils-<x>
```

#### Note

Ubuntu 20 changed the requirements for accepting certificates.

- If you want to install the PEM agent on a machine with an old version of `sslutils`, then you must upgrade `sslutils` to 1.3. Version 1.3 has a 4096-bit RSA key and sha256 signature algorithm support added to it.
- If you don't upgrade `sslutils` to 1.3, then PEM agent might fail to connect to the PEM backend database server, and it might log the error "ca md too weak."

4. If you're using a firewall, allow access to port 8443 on the server where the PEM web application will be located:

```
iptables -t filter -A INPUT -p TCP --dport 8443 -j ACCEPT
```

5. Make sure the components Postgres Enterprise Manager depends on are up to date on all servers. You can do this by updating the whole system using your package manager as shown below. If you prefer to update individual packages, a full list of dependencies is provided in [Dependencies of the PEM Server and Agent on Linux](#).

```
sudo apt-get update
```

## Install the package

```
sudo apt-get -y install edb-pem
```

## Initial configuration

# You can configure the PEM server using the following command:

```
sudo /usr/edb/pem/bin/configure-pem-server.sh
```

For more details, see [Configuring the PEM server on Linux](#).

#### Note

- The operating system user `pem` is created while installing the PEM server. The PEM server web application is a WSGI application, which runs under Apache HTTPD. The `pem` application data and the session is saved to this user's home directory.

## Supported locales

Currently, the Postgres Enterprise Manager server and web interface support a locale of `English(US) en_US` and use of a period (.) as a language separator character. Using an alternate locale or a separator character other than a period might cause errors.

## 7.3.9 Installing Postgres Enterprise Manager server on Debian 11 x86\_64

You can install PEM on a single server, or you can install the web application server and the backend database on two separate servers. You must prepare your servers for PEM installation.

After fulfilling the prerequisites and completing the installation procedure described in the following steps, you must [configure](#) PEM. If you're using two servers, install and configure PEM on both servers.

### Prerequisites

Before you begin the installation process:

1. Install a [supported Postgres instance](#) for PEM to use as a backend database.

You can install this instance on the same server to be used for the PEM web application or on a separate server. You can also use an existing Postgres instance if it is configured as detailed in the next steps.

2. Configure authentication on the Postgres backend database by updating the `pg_hba.conf` file.

Make the following changes manually, prior to configuration. (Additional changes are necessary during [configuration](#).)

- To create the relations required for PEM, the PEM configuration script connects to the Postgres backend database as a superuser of your choice using password authentication. This requires you to permit your chosen superuser to authenticate using a password. This user must be able to connect from any location where you run the configuration script. In practice, this means the server where the backend database is located and the server where the PEM web application is to be installed, if they're different.
- To allow the chosen superuser to connect using password authentication, add a line to `pg_hba.conf` that allows `host` connections using `md5` or `scram-sha-256` authentication, such as `host all superusername 127.0.0.1/32 scram-sha-256`.

#### Note

If you're using EDB Postgres Advanced Server, see [Modifying the pg\\_hba.conf file](#).

If you're using PostgreSQL, see [Client Authentication](#).

3. Verify that the `sslutils` extension is installed on your Postgres server.

If you're using PostgreSQL or EDB Postgres Extended Server on RHEL/AlmaLinux/Rocky Linux or SLES, you also need to install the `hstore contrib` module.

- If you're using EDB Postgres Advanced Server, you can install the `sslutils` extension as follows, where `<x>` is the EDB Postgres Advanced server version.

```
sudo apt-get install edb-as<x>-server-sslutils
```

- If you're using PostgreSQL, you can install the `sslutils` and, if required, `hstore` modules as follows, where `<x>` is the PostgreSQL version.

```
sudo apt-get install postgresql-<x>-sslutils
```

- If you're using EDB Postgres Extended Server, you can install the `sslutils` and, if required, `hstore` modules as follows, where `<x>` is the EDB Postgres Extended Server version.

```
sudo apt-get install edb-postgresextended-sslutils-<x>
```

#### Note

Debian 10 changed the requirements for accepting certificates.

- If you want to install the PEM agent on a machine with an old version of `sslutils`, then you must upgrade `sslutils` to 1.3. Version 1.3 has a 4096-bit RSA key and sha256 signature algorithm support added to it.
- If you don't upgrade `sslutils` to 1.3, then PEM agent might fail to connect to the PEM backend database server, and it might log the error "ca md too weak"

4. If you're using a firewall, allow access to port 8443 on the server where the PEM web application will be located:

```
iptables -t filter -A INPUT -p TCP --dport 8443 -j ACCEPT
```

5. Make sure the components Postgres Enterprise Manager depends on are up to date on all servers. You can do this by updating the whole system using your package manager as shown below. If you prefer to update individual packages, a full list of dependencies is provided in [Dependencies of the PEM Server and Agent on Linux](#).

```
sudo apt-get update
```

## Install the package

```
sudo apt-get -y install edb-pem
```

## Initial configuration

```
# You can configure the PEM server using the following command:
sudo /usr/edb/pem/bin/configure-pem-server.sh
```

For more details, see [Configuring the PEM server on Linux](#).

#### Note

- The operating system user `pem` is created while installing the PEM server. The PEM server web application is a WSGI application, which runs under Apache HTTPD. The `pem` application data and the session is saved to this user's home directory.

## Supported locales

Currently, the Postgres Enterprise Manager server and web interface support a locale of `English(US) en_US` and use of a period (.) as a language separator character. Using an alternate locale or a separator character other than a period might cause errors.

## 7.3.10 Installing Postgres Enterprise Manager server on CentOS 7 x86\_64

You can install PEM on a single server, or you can install the web application server and the backend database on two separate servers. You must prepare your servers for PEM installation.

After fulfilling the prerequisites and completing the installation procedure described in the following steps, you must [configure](#) PEM. If you're using two servers, install and configure PEM on both servers.

### Prerequisites

Before you begin the installation process:

1. Install a [supported Postgres instance](#) for PEM to use as a backend database.

You can install this instance on the same server to be used for the PEM web application or on a separate server. You can also use an existing Postgres instance if it is configured as detailed in the next steps.

2. Configure authentication on the Postgres backend database by updating the `pg_hba.conf` file.

Make the following changes manually, prior to configuration. (Additional changes are necessary during [configuration](#).)

- To create the relations required for PEM, the PEM configuration script connects to the Postgres backend database as a superuser of your choice using password authentication. This requires you to permit your chosen superuser to authenticate using a password. This user must be able to connect from any location where you run the configuration script. In practice, this means the server where the backend database is located and the server where the PEM web application is to be installed, if they're different.
- To allow the chosen superuser to connect using password authentication, add a line to `pg_hba.conf` that allows `host` connections using `md5` or `scram-sha-256` authentication, such as `host all superusername 127.0.0.1/32 scram-sha-256`.

#### Note

If you're using EDB Postgres Advanced Server, see [Modifying the pg\\_hba.conf file](#).

If you're using PostgreSQL, see [Client Authentication](#).

3. Verify that the `sslutils` extension is installed on your Postgres server.

If you're using PostgreSQL or EDB Postgres Extended Server on RHEL/AlmaLinux/Rocky Linux or SLES, you also need to install the `hstore contrib` module.

- If you're using EDB Postgres Advanced Server, you can install the `sslutils` extension as follows, where `<x>` is the EDB Postgres Advanced server version.

```
sudo yum install edb-as<x>-server-sslutils
```

- If you're using PostgreSQL, you can install the `sslutils` and, if required, `hstore` modules as follows, where `<x>` is the PostgreSQL version.

```
sudo yum install sslutils_<x> postgresql<x>-contrib
```

- If you're using EDB Postgres Extended Server, you can install the `sslutils` and, if required, `hstore` modules as follows, where `<x>` is the EDB Postgres Extended Server version.

```
sudo yum install edb-postgresextened<x>-sslutils edb-postgresextened<x>-contrib
```

4. If you're using a firewall, allow access to port 8443 on the server where the PEM web application will be located:

```
firewall-cmd --permanent --zone=public --add-port=8443/tcp
firewall-cmd --reload
```

5. Make sure the components Postgres Enterprise Manager depends on are up to date on all servers. You can do this by updating the whole system using your package manager as shown below. If you prefer to update individual packages, a full list of dependencies is provided in [Dependencies of the PEM Server and Agent on Linux](#).

```
sudo yum upgrade
```

## Install the package

```
sudo yum -y install edb-pem
```

## Initial configuration

```
# You can configure the PEM server using the following command:
sudo /usr/edb/pem/bin/configure-pem-server.sh
```

For more details, see [Configuring the PEM server on Linux](#).

### Note

- The operating system user pem is created while installing the PEM server. The PEM server web application is a WSGI application, which runs under Apache HTTPD. The pem application data and the session is saved to this user's home directory.

## Supported locales

Currently, the Postgres Enterprise Manager server and web interface support a locale of **English(US) en\_US** and use of a period (.) as a language separator character. Using an alternate locale or a separator character other than a period might cause errors.

- If you're doing a fresh installation of the PEM server on a CentOS 7.x host, the installer also installs edb-python3-mod\_wsgi packages with the installation required by the operating system.
- If you're upgrading the PEM server on a CentOS 7.x host, the mod\_wsgi system package is replaced by the edb-python3-mod\_wsgi package as required by the operating system.

### 7.3.11 Installing Postgres Enterprise Manager server on Debian 10 x86\_64

You can install PEM on a single server, or you can install the web application server and the backend database on two separate servers. You must prepare

your servers for PEM installation.

After fulfilling the prerequisites and completing the installation procedure described in the following steps, you must [configure](#) PEM. If you're using two servers, install and configure PEM on both servers.

## Prerequisites

Before you begin the installation process:

1. Install a [supported Postgres instance](#) for PEM to use as a backend database.

You can install this instance on the same server to be used for the PEM web application or on a separate server. You can also use an existing Postgres instance if it is configured as detailed in the next steps.

2. Configure authentication on the Postgres backend database by updating the `pg_hba.conf` file.

Make the following changes manually, prior to configuration. (Additional changes are necessary during [configuration](#).)

- To create the relations required for PEM, the PEM configuration script connects to the Postgres backend database as a superuser of your choice using password authentication. This requires you to permit your chosen superuser to authenticate using a password. This user must be able to connect from any location where you run the configuration script. In practice, this means the server where the backend database is located and the server where the PEM web application is to be installed, if they're different.
- To allow the chosen superuser to connect using password authentication, add a line to `pg_hba.conf` that allows `host` connections using `md5` or `scram-sha-256` authentication, such as `host all superusername 127.0.0.1/32 scram-sha-256`.

### Note

If you're using EDB Postgres Advanced Server, see [Modifying the pg\\_hba.conf file](#).

If you're using PostgreSQL, see [Client Authentication](#).

3. Verify that the `sslutils` extension is installed on your Postgres server.

If you're using PostgreSQL or EDB Postgres Extended Server on RHEL/AlmaLinux/Rocky Linux or SLES, you also need to install the `hstore contrib` module.

- If you're using EDB Postgres Advanced Server, you can install the `sslutils` extension as follows, where `<x>` is the EDB Postgres Advanced server version.

```
sudo apt-get install edb-as<x>-server-sslutils
```

- If you're using PostgreSQL, you can install the `sslutils` and, if required, `hstore` modules as follows, where `<x>` is the PostgreSQL version.

```
sudo apt-get install postgresql-<x>-sslutils
```

- If you're using EDB Postgres Extended Server, you can install the `sslutils` and, if required, `hstore` modules as follows, where `<x>` is the EDB Postgres Extended Server version.

```
sudo apt-get install edb-postgresextended-sslutils-<x>
```

### Note

Debian 10 changed the requirements for accepting certificates.

- If you want to install the PEM agent on a machine with an old version of sslutils, then you must upgrade sslutils to 1.3. Version 1.3 has a 4096-bit RSA key and sha256 signature algorithm support added to it.
- If you don't upgrade sslutils to 1.3, then PEM agent might fail to connect to the PEM backend database server, and it might log the error "ca md too weak"

4. If you're using a firewall, allow access to port 8443 on the server where the PEM web application will be located:

```
iptables -t filter -A INPUT -p TCP --dport 8443 -j ACCEPT
```

5. Make sure the components Postgres Enterprise Manager depends on are up to date on all servers. You can do this by updating the whole system using your package manager as shown below. If you prefer to update individual packages, a full list of dependencies is provided in [Dependencies of the PEM Server and Agent on Linux](#).

```
sudo apt-get update
```

## Install the package

```
sudo apt-get -y install edb-pem
```

## Initial configuration

# You can configure the PEM server using the following command:  

```
sudo /usr/edb/pem/bin/configure-pem-server.sh
```

For more details, see [Configuring the PEM server on Linux](#).

### Note

- The operating system user pem is created while installing the PEM server. The PEM server web application is a WSGI application, which runs under Apache HTTPD. The pem application data and the session is saved to this user's home directory.

## Supported locales

Currently, the Postgres Enterprise Manager server and web interface support a locale of `English(US) en_US` and use of a period (.) as a language separator character. Using an alternate locale or a separator character other than a period might cause errors.

## 7.3.12 Installing Postgres Enterprise Manager server on RHEL 7 or OL 7 x86\_64

You can install PEM on a single server, or you can install the web application server and the backend database on two separate servers. You must prepare your servers for PEM installation.

After fulfilling the prerequisites and completing the installation procedure described in the following steps, you must [configure](#) PEM. If you're using two servers, install and configure PEM on both servers.

## Prerequisites

Before you begin the installation process:

1. Install a [supported Postgres instance](#) for PEM to use as a backend database.

You can install this instance on the same server to be used for the PEM web application or on a separate server. You can also use an existing Postgres instance if it is configured as detailed in the next steps.

2. Configure authentication on the Postgres backend database by updating the `pg_hba.conf` file.

Make the following changes manually, prior to configuration. (Additional changes are necessary during [configuration](#).)

- To create the relations required for PEM, the PEM configuration script connects to the Postgres backend database as a superuser of your choice using password authentication. This requires you to permit your chosen superuser to authenticate using a password. This user must be able to connect from any location where you run the configuration script. In practice, this means the server where the backend database is located and the server where the PEM web application is to be installed, if they're different.
- To allow the chosen superuser to connect using password authentication, add a line to `pg_hba.conf` that allows `host` connections using `md5` or `scram-sha-256` authentication, such as `host all superusername 127.0.0.1/32 scram-sha-256`.

### Note

If you're using EDB Postgres Advanced Server, see [Modifying the pg\\_hba.conf file](#).

If you're using PostgreSQL, see [Client Authentication](#).

3. Verify that the `sslutils` extension is installed on your Postgres server.

If you're using PostgreSQL or EDB Postgres Extended Server on RHEL/AlmaLinux/Rocky Linux or SLES, you also need to install the `hstore contrib` module.

- If you're using EDB Postgres Advanced Server, you can install the `sslutils` extension as follows, where `<x>` is the EDB Postgres Advanced server version.

```
sudo yum install edb-as<x>-server-sslutils
```

- If you're using PostgreSQL, you can install the `sslutils` and, if required, `hstore` modules as follows, where `<x>` is the PostgreSQL version.

```
sudo yum install sslutils_<x> postgresql<x>-contrib
```

- If you're using EDB Postgres Extended Server, you can install the `sslutils` and, if required, `hstore` modules as follows, where `<x>` is the EDB Postgres Extended Server version.

```
sudo yum install edb-postgresextended<x>-sslutils edb-postgresextended<x>-contrib
```

4. If you're using a firewall, allow access to port 8443 on the server where the PEM web application will be located:

```
firewall-cmd --permanent --zone=public --add-port=8443/tcp
```

```
firewall-cmd --reload
```

5. Make sure the components Postgres Enterprise Manager depends on are up to date on all servers. You can do this by updating the whole system



using your package manager as shown below. If you prefer to update individual packages, a full list of dependencies is provided in [Dependencies of the PEM Server and Agent on Linux](#).

```
sudo yum upgrade
```

## Install the package

```
sudo yum -y install edb-pem
```

## Initial configuration

```
# You can configure the PEM server using the following command:  
sudo /usr/edb/pem/bin/configure-pem-server.sh
```

For more details, see [Configuring the PEM server on Linux](#).

### Note

- The operating system user pem is created while installing the PEM server. The PEM server web application is a WSGI application, which runs under Apache HTTPD. The pem application data and the session is saved to this user's home directory.

## Supported locales

Currently, the Postgres Enterprise Manager server and web interface support a locale of `English(US) en_US` and use of a period (.) as a language separator character. Using an alternate locale or a separator character other than a period might cause errors.

- If you're doing a fresh installation of the PEM server on a RHEL 7.x host, the installer also installs edb-python3-mod\_wsgi packages with the installation required by the operating system.
- If you're upgrading the PEM server on a RHEL 7.x host, the mod\_wsgi system package is replaced by the edb-python3-mod\_wsgi package as required by the operating system.

## 7.4 Installing Postgres Enterprise Manager server on Linux IBM Power (ppc64le)

Operating system-specific install instructions are described in the corresponding documentation:

### Red Hat Enterprise Linux (RHEL)

- [RHEL 9](#)
- [RHEL 8](#)

## SUSE Linux Enterprise (SLES)

- [SLES 15](#)
- [SLES 12](#)

### 7.4.1 Installing Postgres Enterprise Manager server on RHEL 9 ppc64le

You can install PEM on a single server, or you can install the web application server and the backend database on two separate servers. You must prepare your servers for PEM installation.

After fulfilling the prerequisites and completing the installation procedure described in the following steps, you must [configure](#) PEM. If you're using two servers, install and configure PEM on both servers.

#### Prerequisites

Before you begin the installation process:

1. Install a [supported Postgres instance](#) for PEM to use as a backend database.

You can install this instance on the same server to be used for the PEM web application or on a separate server. You can also use an existing Postgres instance if it is configured as detailed in the next steps.

2. Configure authentication on the Postgres backend database by updating the `pg_hba.conf` file.

Make the following changes manually, prior to configuration. (Additional changes are necessary during [configuration](#).)

- To create the relations required for PEM, the PEM configuration script connects to the Postgres backend database as a superuser of your choice using password authentication. This requires you to permit your chosen superuser to authenticate using a password. This user must be able to connect from any location where you run the configuration script. In practice, this means the server where the backend database is located and the server where the PEM web application is to be installed, if they're different.
- To allow the chosen superuser to connect using password authentication, add a line to `pg_hba.conf` that allows `host` connections using `md5` or `scram-sha-256` authentication, such as `host all superusername 127.0.0.1/32 scram-sha-256`.

#### Note

If you're using EDB Postgres Advanced Server, see [Modifying the pg\\_hba.conf file](#).

If you're using PostgreSQL, see [Client Authentication](#).

3. Verify that the `sslutils` extension is installed on your Postgres server.

If you're using PostgreSQL or EDB Postgres Extended Server on RHEL/AlmaLinux/Rocky Linux or SLES, you also need to install the `hstore contrib` module.

- If you're using EDB Postgres Advanced Server, you can install the `sslutils` extension as follows, where `<x>` is the EDB Postgres Advanced server version.

```
sudo dnf install edb-as<x>-server-sslutils
```

- If you're using PostgreSQL, you can install the `sslutils` and, if required, `hstore` modules as follows, where `<x>` is the PostgreSQL version.

```
sudo dnf install sslutils_<x> postgresql<x>-contrib
```

- If you're using EDB Postgres Extended Server, you can install the `sslutils` and, if required, `hstore` modules as follows, where `<x>` is the EDB Postgres Extended Server version.

```
sudo dnf install edb-postgresextended<x>-sslutils edb-postgresextended<x>-contrib
```

4. If you're using a firewall, allow access to port 8443 on the server where the PEM web application will be located:

```
firewall-cmd --permanent --zone=public --add-port=8443/tcp
```

```
firewall-cmd --reload
```

5. Make sure the components Postgres Enterprise Manager depends on are up to date on all servers. You can do this by updating the whole system using your package manager as shown below. If you prefer to update individual packages, a full list of dependencies is provided in [Dependencies of the PEM Server and Agent on Linux](#).

```
sudo dnf upgrade
```

## Install the package

```
sudo dnf -y install edb-pem
```

## Initial configuration

```
# You can configure the PEM server using the following command:
sudo /usr/edb/pem/bin/configure-pem-server.sh
```

For more details, see [Configuring the PEM server on Linux](#).

### Note

- The operating system user `pem` is created while installing the PEM server. The PEM server web application is a WSGI application, which runs under Apache HTTPD. The pem application data and the session is saved to this user's home directory.

## Supported locales

Currently, the Postgres Enterprise Manager server and web interface support a locale of `English(US) en_US` and use of a period (.) as a language separator character. Using an alternate locale or a separator character other than a period might cause errors.

## 7.4.2 Installing Postgres Enterprise Manager server on RHEL 8 ppc64le

You can install PEM on a single server, or you can install the web application server and the backend database on two separate servers. You must prepare your servers for PEM installation.

After fulfilling the prerequisites and completing the installation procedure described in the following steps, you must [configure](#) PEM. If you're using two servers, install and configure PEM on both servers.

## Prerequisites

Before you begin the installation process:

1. Install a [supported Postgres instance](#) for PEM to use as a backend database.

You can install this instance on the same server to be used for the PEM web application or on a separate server. You can also use an existing Postgres instance if it is configured as detailed in the next steps.

2. Configure authentication on the Postgres backend database by updating the `pg_hba.conf` file.

Make the following changes manually, prior to configuration. (Additional changes are necessary during [configuration](#).)

- To create the relations required for PEM, the PEM configuration script connects to the Postgres backend database as a superuser of your choice using password authentication. This requires you to permit your chosen superuser to authenticate using a password. This user must be able to connect from any location where you run the configuration script. In practice, this means the server where the backend database is located and the server where the PEM web application is to be installed, if they're different.
- To allow the chosen superuser to connect using password authentication, add a line to `pg_hba.conf` that allows `host` connections using `md5` or `scram-sha-256` authentication, such as `host all superusername 127.0.0.1/32 scram-sha-256`.

### Note

If you're using EDB Postgres Advanced Server, see [Modifying the pg\\_hba.conf file](#).

If you're using PostgreSQL, see [Client Authentication](#).

3. Verify that the `sslutils` extension is installed on your Postgres server.

If you're using PostgreSQL or EDB Postgres Extended Server on RHEL/AlmaLinux/Rocky Linux or SLES, you also need to install the `hstore contrib` module.

- If you're using EDB Postgres Advanced Server, you can install the `sslutils` extension as follows, where `<x>` is the EDB Postgres Advanced server version.

```
sudo dnf install edb-as<x>-server-sslutils
```

- If you're using PostgreSQL, you can install the `sslutils` and, if required, `hstore` modules as follows, where `<x>` is the PostgreSQL version.

```
sudo dnf install sslutils_<x> postgresql<x>-contrib
```

- If you're using EDB Postgres Extended Server, you can install the `sslutils` and, if required, `hstore` modules as follows, where `<x>` is the EDB Postgres Extended Server version.

```
sudo dnf install edb-postgresextended<x>-sslutils edb-postgresextended<x>-contrib
```

4. If you're using a firewall, allow access to port 8443 on the server where the PEM web application will be located:

```
firewall-cmd --permanent --zone=public --add-port=8443/tcp
firewall-cmd --reload
```

5. Make sure the components Postgres Enterprise Manager depends on are up to date on all servers. You can do this by updating the whole system using your package manager as shown below. If you prefer to update individual packages, a full list of dependencies is provided in [Dependencies of the PEM Server and Agent on Linux](#).

```
sudo dnf upgrade
```

## Install the package

```
sudo dnf -y install edb-pem
```

## Initial configuration

```
# You can configure the PEM server using the following command:
sudo /usr/edb/pem/bin/configure-pem-server.sh
```

For more details, see [Configuring the PEM server on Linux](#).

### Note

- The operating system user pem is created while installing the PEM server. The PEM server web application is a WSGI application, which runs under Apache HTTPD. The pem application data and the session is saved to this user's home directory.

## Supported locales

Currently, the Postgres Enterprise Manager server and web interface support a locale of `English(US) en_US` and use of a period (.) as a language separator character. Using an alternate locale or a separator character other than a period might cause errors.

## 7.4.3 Installing Postgres Enterprise Manager server on SLES 15 ppc64le

### Note

Postgres Enterprise Manager 8.3 and later is supported on SLES.

## Prerequisites

Before you begin the installation process:

1. Install a [supported Postgres instance](#) for PEM to use as a backend database.

You can install this instance on the same server to be used for the PEM web application or on a separate server. You can also use an existing Postgres instance if it is configured as detailed in the next steps.

2. Configure authentication on the Postgres backend database by updating the `pg_hba.conf` file.

Make the following changes manually, prior to configuration. (Additional changes are necessary during [configuration](#).)

- To create the relations required for PEM, the PEM configuration script connects to the Postgres backend database as a superuser of your choice using password authentication. This requires you to permit your chosen superuser to authenticate using a password. This user must be able to connect from any location where you run the configuration script. In practice, this means the server where the backend database is located and the server where the PEM web application is to be installed, if they're different.
- To allow the chosen superuser to connect using password authentication, add a line to `pg_hba.conf` that allows `host` connections using `md5` or `scram-sha-256` authentication, such as `host all superusername 127.0.0.1/32 scram-sha-256`.

#### Note

If you're using EDB Postgres Advanced Server, see [Modifying the pg\\_hba.conf file](#).

If you're using PostgreSQL, see [Client Authentication](#).

3. Verify that the `sslutils` extension is installed on your Postgres server.

If you're using PostgreSQL or EDB Postgres Extended Server on RHEL/AlmaLinux/Rocky Linux or SLES, you also need to install the `hstore contrib` module.

- If you're using EDB Postgres Advanced Server, you can install the `sslutils` extension as follows, where `<x>` is the EDB Postgres Advanced server version.

```
sudo zypper install edb-as<x>-server-sslutils
```

- If you're using PostgreSQL, you can install the `sslutils` and, if required, `hstore` modules as follows, where `<x>` is the PostgreSQL version.

```
sudo zypper install sslutils_<x> postgresql<x>-contrib
```

- If you're using EDB Postgres Extended Server, you can install the `sslutils` and, if required, `hstore` modules as follows, where `<x>` is the EDB Postgres Extended Server version.

```
sudo zypper install edb-postgresextended<x>-sslutils edb-postgresextended<x>-contrib
```

4. If you're using a firewall, allow access to port 8443 on the server where the PEM web application will be located:

```
firewall-cmd --permanent --zone=public --add-port=8443/tcp
firewall-cmd --reload
```

5. Make sure the components Postgres Enterprise Manager depends on are up to date on all servers. You can do this by updating the whole system using your package manager as shown below. If you prefer to update individual packages, a full list of dependencies is provided in [Dependencies of the PEM Server and Agent on Linux](#).

```
sudo zypper update
```

## Install the package

```
sudo zypper -n install edb-pem
```

## Initial configuration

```
# You can configure the PEM server using the following command:  
sudo /usr/edb/pem/bin/configure-pem-server.sh
```

For more details, see [Configuring the PEM server on Linux](#).

### Note

- The operating system user pem is created while installing the PEM server. The PEM server web application is a WSGI application, which runs under Apache HTTPD. The pem application data and the session is saved to this user's home directory.

## Supported locales

Currently, the Postgres Enterprise Manager server and web interface support a locale of `English(US) en_US` and use of a period (.) as a language separator character. Using an alternate locale or a separator character other than a period might cause errors.

## 7.4.4 Installing Postgres Enterprise Manager server on SLES 12 ppc64le

### Note

Postgres Enterprise Manager 8.3 and later is supported on SLES.

## Prerequisites

Before you begin the installation process:

1. Install a [supported Postgres instance](#) for PEM to use as a backend database.

You can install this instance on the same server to be used for the PEM web application or on a separate server. You can also use an existing Postgres instance if it is configured as detailed in the next steps.

2. Configure authentication on the Postgres backend database by updating the `pg_hba.conf` file.

Make the following changes manually, prior to configuration. (Additional changes are necessary during [configuration](#).)

- To create the relations required for PEM, the PEM configuration script connects to the Postgres backend database as a superuser of your choice using password authentication. This requires you to permit your chosen superuser to authenticate using a password. This user must be able to connect from any location where you run the configuration script. In practice, this means the server where the backend database is located and the server where the PEM web application is to be installed, if they're different.

- To allow the chosen superuser to connect using password authentication, add a line to `pg_hba.conf` that allows `host` connections using `md5` or `scram-sha-256` authentication, such as `host all superusername 127.0.0.1/32 scram-sha-256`.

#### Note

If you're using EDB Postgres Advanced Server, see [Modifying the `pg\_hba.conf` file](#).

If you're using PostgreSQL, see [Client Authentication](#).

3. Verify that the `sslutils` extension is installed on your Postgres server.

If you're using PostgreSQL or EDB Postgres Extended Server on RHEL/AlmaLinux/Rocky Linux or SLES, you also need to install the `hstore contrib` module.

- If you're using EDB Postgres Advanced Server, you can install the `sslutils` extension as follows, where `<x>` is the EDB Postgres Advanced server version.

```
sudo zypper install edb-as<x>-server-sslutils
```

- If you're using PostgreSQL, you can install the `sslutils` and, if required, `hstore` modules as follows, where `<x>` is the PostgreSQL version.

```
sudo zypper install sslutils_<x> postgresql<x>-contrib
```

- If you're using EDB Postgres Extended Server, you can install the `sslutils` and, if required, `hstore` modules as follows, where `<x>` is the EDB Postgres Extended Server version.

```
sudo zypper install edb-postgresextended<x>-sslutils edb-postgresextended<x>-contrib
```

4. If you're using a firewall, allow access to port 8443 on the server where the PEM web application will be located:

```
firewall-cmd --permanent --zone=public --add-port=8443/tcp
```

```
firewall-cmd --reload
```

5. Make sure the components Postgres Enterprise Manager depends on are up to date on all servers. You can do this by updating the whole system using your package manager as shown below. If you prefer to update individual packages, a full list of dependencies is provided in [Dependencies of the PEM Server and Agent on Linux](#).

```
sudo zypper update
```

## Install the package

```
sudo zypper -n install edb-pem
```

## Initial configuration

```
# You can configure the PEM server using the following command:
```



```
sudo /usr/edb/pem/bin/configure-pem-server.sh
```

For more details, see [Configuring the PEM server on Linux](#).

#### Note

- The operating system user pem is created while installing the PEM server. The PEM server web application is a WSGI application, which runs under Apache HTTPD. The pem application data and the session is saved to this user's home directory.

## Supported locales

Currently, the Postgres Enterprise Manager server and web interface support a locale of `English(US) en_US` and use of a period (.) as a language separator character. Using an alternate locale or a separator character other than a period might cause errors.

## 7.5 Installing Postgres Enterprise Manager server on Windows

On Windows, the PEM server graphical installer installs and configures the PEM server, a PEM agent, and the software required to connect to the PEM web interface with your choice of browser.

At the heart of each PEM installation is the PEM server. In a production environment, the server typically is a dedicated machine, monitoring a large number of Postgres servers or a smaller number of busy servers.

The PEM server backend database can be an EDB distribution of the PostgreSQL or EDB Postgres Advanced Server database server or an existing Postgres server installed from another source. The Postgres backend database server must be version 11 or later and contains a database named `pem`, which is used by the PEM server as a repository.

- If you want to use an existing Postgres server to host the PEM server, the PEM server installer can create the `pem` database on the Postgres host. You must manually satisfy the software prerequisites if you choose to use an existing server.

For more information about using an existing Postgres server to host the PEM server backend database, see [Installing the PEM server on an existing Postgres server](#).

- If you don't want to use an existing installation of Postgres as the PEM server host, the PEM server installer can:
  - Install PostgreSQL
  - Satisfy the server host's software prerequisites
  - Create an instance (a PostgreSQL database cluster) that contains the `pem` database, which is the simplest PEM server installation option
- PEM-HTTPD is made available for Postgres installations through the PEM server installer or the StackBuilder utility. If PEM-HTTPD is already installed on the host, the PEM server installer reviews and updates the existing installation if needed. If the PEM server host doesn't contain an existing PEM-HTTPD installation, the PEM server installer adds it.
- Before installing the PEM server, you must decide if you want to run PostgreSQL and PEM-HTTPD on the same host or on separate hosts. If you intend to run the PostgreSQL database server and PEM-HTTPD on different hosts, then you must run the PEM server installer twice, once on each host. See [Installing the PEM server and PEM-HTTPD on separate hosts](#).
- For detailed information about installing the PEM server and PEM-HTTPD on the same host, see [Installing the PEM server and PEM-HTTPD on the same host](#).
- For detailed information about installing and configuring a standalone PEM agent, see [Installing the PEM agent on Windows](#).
- Language pack installers contain supported languages that you can use with EDB Postgres Advanced Server and EDB PostgreSQL database

installers. The language pack installer allows you to install Perl, TCL/TK, and Python without installing supporting software from third-party vendors. For more information about installing and using the language pack, see [EDB Postgres language pack](#).

- For troubleshooting the installation or configuration of the PEM agent, see [Troubleshooting PEM agent](#).

The PEM server installer also installs the software required to access the server using the PEM web interface. You can access the web interface with a supported version of your browser. You can use the web interface to:

- Review information about objects that reside on monitored servers
- Manage databases and database objects that reside on monitored servers
- Review statistical information gathered by the PEM server

#### Note

If you're using SSL certificates, make sure that all the SSL certificates are in the data directory in the backend database server. If the certificates aren't in the data directory, then the PEM server's configuration might fail because it looks in the data directory while configuring the PEM server.

## 7.5.1 Installing the PEM server and PEM-HTTPD on the same host

The easiest PEM server installation configuration consists of a PEM backend database server (hosted on a PostgreSQL database installed with the PEM server installer) and a PEM-HTTPD service that reside on the same host. In this configuration, the PEM server installer provides the prerequisite software for the backend host to register the service (on Windows).

1. Invoke the PEM server installer. On a Windows system, right-click the installer icon and select **Run as Administrator**. The installer displays a Welcome screen. Select **Next**.
2. Review the license agreement before selecting the appropriate radio button and accepting the agreement. Select **Next**.
3. Use the Installation Directory dialog box to specify the location of the PEM server:
  - By default, the PEM server is installed in `C:\Program Files\edb\pem` on Windows. Accept the default location, or use the **Installation Directory** button to open a dialog box and select the directory to install the PEM server in.
  - Use the **Show advanced options** check box to open the Advanced Options dialog box during installation. Use the Advanced Options dialog box when installing the Postgres database server and the PEM-HTTPD on different hosts or if you want the PEM server to reside on an existing Postgres server installation.
  - To install the PostgreSQL server package with the installer and PEM-HTTPD on the same host, clear **Show advanced options** and select **Next**.

The PEM server installer performs a preinstallation check for PEM-HTTPD, Language Pack, and PostgreSQL 13. If the installer doesn't locate these packages, it informs you in the Dependency Missing dialog box.

#### Note

By default EDB Language Pack is installed in `C:\edb\languagepack\v1`.

1. If the dependencies are missing, the PEM server installer launches the respective installation wizards. Follow the wizard's onscreen directions each package.

After installing any missing dependencies, the installation process continues by displaying the Database Server Installation Details dialog box. The information provided on the Database Server Installation Details dialog box enables the installer to connect to the PostgreSQL server.

2. Provide the user name and password of a database superuser. After supplying the requested information, select **Next**.
3. After providing the name and password of the Postgres database superuser, you might be prompted for the password to the user account under

which the PEM agent will run. If prompted, provide the password, and select **Next**.

4. Use the Network Details dialog box to specify the CIDR-style network address from which the PEM agents will connect to the server (the *client-side* address).

You can specify the address of a network host or a network address range. For example, if you want to monitor database servers with the addresses `192.168.10.23`, `192.168.10.76`, and `192.168.10.184`, enter `192.168.10.0/24` to allow connections with hosts in that network.

The specified address is added to the server's `pg_hba.conf` file. You can specify more network addresses by manually adding entries to the `pg_hba.conf` file on the PostgreSQL server. Use the first entry as a template.

After you add the network address, select **Next**.

The PEM server installer installs a PEM agent on the host where the server resides to monitor the server and provide alert processing and garbage collection services. A certificate is also installed in the location specified in the **Agent certificate path** field.

5. Enter an alternate description or select an alternate agent certificate path for the PEM agent, or accept the defaults. Select **Next**.
6. The wizard is now ready to install the PEM server. Select **Next** to continue with the installation.

During the installation process, the installer copies files to the system and sets up the database and web services required to run PEM. When the installation completes, a confirmation indicates that:

- The web service was configured.
- The web service is listening on port 8443.
- The `pem` database was created and configured.

7. Select **OK**.

## 7.5.2 Installing the PEM server and PEM-HTTPD on separate hosts

To use separate hosts for the PEM server backend database and PEM-HTTPD:

1. Invoke the PEM server installer on the host of the Postgres server that will contain the `pem` database. During the installation, select the **Database** option on the Advanced Options dialog box, and provide connection information for the Postgres server.
2. Modify the `pg_hba.conf` file of the Postgres installation on which the PEM server (and `pem` database) resides, allowing connections from the host of the PEM-HTTPD server.
3. Invoke the PEM server installer on the host of the PEM-HTTPD server, selecting the **Web Services** option on the Installation Type dialog box.

To start the installation:

1. On a Windows system, invoke the PEM server installer. Right-click the installer and select **Run as Administrator**. The installer displays a Welcome screen. Select **Next**.
2. Review the license agreement before selecting the appropriate radio button and accepting the agreement. Select **Next**.
3. Use fields on the Installation Directory dialog box to specify the directory for the PEM server to reside and to open the Advanced Options dialog box during installation:
  - By default, the PEM server is installed in `C:\Program Files\edb\pem` on Windows. Accept the default location, or use the **Installation Directory** field to open a browser dialog box and select the directory to install the PEM server in.
  - To install the PEM server and PEM-HTTPD on separate hosts, use the Advanced Options dialog box to specify the installation type (**Web Services** or **Database**). Select **Show advanced options** to include the Advanced Options dialog box in the installation process.
  - Select **Next**.
4. Use the Advanced Options dialog box to specify the components that you want to install:

- Select **Web Services and Database** to indicate that the Postgres server and PEM-HTTPD will both reside on the current host. If you select **Web Services and Database**, the PEM server installer lets you specify the Postgres server to use for the PEM server before checking for a PEM-HTTPD installation.
- Select **Web Services** to install PEM-HTTPD on the current host, while using a Postgres database server that resides on another host to host the PEM server and `pem` database.

#### Note

Before using the **Web Services** option to install PEM-HTTPD, you must complete the PEM server installation process on the host of the PEM server and `pem` backend database. Select **Database** on the Advanced Options dialog box and modify the connection properties of the `pg_hba.conf` file on the PEM server.

This option invokes the installation steps described in [Installing web services](#).

- Select **Database** to use an existing Postgres server (version 11 or later) or to install only the database server that's distributed with the PEM server installer. This option invokes the installation steps described in [Specifying a database host](#).

5. After selecting an installation option, select **Next**.

## Specifying a database host

1. Select **Database** on the Advanced Options dialog box to specify connection information for the host where the PEM server backend database (named `pem`) will reside. Select **Next**.
2. Use the list on the Database Server Selection dialog box to select a host for the PEM server backend database. You can:
  - Select a host from existing Postgres installations that reside on the current host.

#### Note

You might need to add the `sslutils` package to your installation.

- Select **PostgreSQL <x>** to install the Postgres server that's distributed with the PEM server installer, where `<x>` is the PostgreSQL database server version. If you decide to use the version of PostgreSQL that's bundled with the PEM server installer, the EnterpriseDB one-click PostgreSQL installer opens and walks you through the installation.
- Select **Other Database Server** to specify connection information for a Postgres server that wasn't installed using a one-click graphical installer from EnterpriseDB. For information about the software prerequisites for the PEM server database host, see [Preparing the Postgres server](#).
- Select **Next**.

If the PEM server will reside on an existing Postgres server, the Database Server Installation Details dialog box opens.

3. The information required on the Database Server Installation Details dialog box can vary. The PEM server installer asks you to provide only the information about the selected installation that it can't locate:
  - Specify the name of a Postgres database superuser in the **User** field.
  - Specify the password associated with that user in the **Password** field.
  - Select **Next**.
4. If prompted, provide the system password for the service account under which the PEM agent will run. Select **Next**.
5. Use the Network Details dialog box to specify the CIDR-style network address from which PEM agents will connect to the server (the client-side address). The specified address is added to the server's `pg_hba.conf` file.

**Note**

You can specify more network addresses by manually adding entries to the `pg_hba.conf` file on the PostgreSQL server.

Accept the default (specifying the localhost), or specify a **Network address** range. Select **Next**.

The PEM server installer installs a PEM agent on the host on which the server resides to monitor the server and provide alert processing and garbage collection services. A certificate is also installed in the location specified in the **Agent certificate path** field.

6. You can enter an alternate description or an alternate agent certificate path for the PEM agent or accept the defaults. Select **Next**.

7. The wizard is now ready to install the PEM server. Select **Next** to proceed with the installation.

During the installation process, the installer copies files to the system and sets up the PEM server's backend database. A confirmation indicates that the `pem` database was created and configured.

8. Select **OK**

When the database portion of the PEM server installation is complete, you can invoke the PEM server on another host to install or upgrade PEM-HTTPD.

## Installing web services

Select **Web Services** on the Advanced Options dialog box to either install PEM-HTTPD on the current host or update an existing PEM-HTTPD installation.

**Note**

The current host might not be the host of the PEM backing database.

Before selecting this option, you must:

- Install the PEM server installer on a host system, during which you specify a backing database for the PEM server.
- Modify the `pg_hba.conf` file on the PEM server database host to allow connections from the PEM-HTTPD host.
- Restart the database server.

1. Select **Web Services** and select **Next**. The PEM server installer checks the current host for existing PEM-HTTPD and LanguagePack installations.
2. If the installer doesn't locate the components, the installer informs you that one or more dependencies are missing. Select **Next** to install the language pack. After installing language pack, the installer invokes the PEM-HTTPD setup wizard.
3. Follow the onscreen instructions of the PEM-HTTPD Setup wizard. When the wizard completes the installation, select **Finish** to open the Database Server Installation Details dialog box.
4. Use the Database Server Installation Details dialog box to provide connection information for the Postgres installation that's hosting the PEM server installation:
  - Enter the name or IP address of the PEM server host in the **Host** field.
  - Enter the port number on which the Postgres server is listening in the **Port** field.
  - Enter the name of a Postgres database superuser in the **User** field.
  - Enter the password associated with the Postgres superuser in the **Password** field.
5. Select **Next**.

Before completing the PEM server installation, the installer contacts the database host. For the installation to continue, the `pg_hba.conf` file on the PEM database host must be configured to accept connections from the host of the httpd server, and the firewall must allow a connection. The

PEM server installer completes the PEM server installation, adding only those items that must reside on the host of the PEM-HTTPD server.

### 7.5.3 Installing the PEM server on an existing Postgres servers

You can use an existing Postgres server (version 11 or later) to host the PEM server and the `pem` database. Postgres installers and prerequisite software extensions are freely available on the [EnterpriseDB website](#).

You can configure an existing Postgres server for a PEM server installation.

#### Note

These instructions are guidelines. the actual steps needed to configure your Postgres installation depend on the configuration of your Postgres server.

The following versions of Postgres are preconfigured to contain the `sslutils` extension and a service script. No additional preparation is required to use the following Postgres versions as a PEM backend database server:

- PostgreSQL 11 or later (as bundled with the PEM Server installer)
- EDB Postgres Advanced Server 11 or later

#### Preparing the Postgres server

Before installing the PEM server on an existing Postgres server, you must:

- Ensure that the Postgres server contains an installation of the `sslutils` extension. For more information, see [Installing the sslutils Extension](#).
- Register the server with the Windows service manager. For more information, see [Registering the service](#).

After preparing the server, you can use the PEM server installer to install PEM on the existing Postgres server.

#### Installing the sslutils extension

The Postgres server where the PEM server will reside must contain the `sslutils` extension. The `sslutils` package is freely available for download from the [EDB website](#).

When the web page opens, select the link for the `SRC- SSL Utils 1.3` package. When the download completes, extract the file, and copy it into the Postgres installation directory.

#### Note

You don't need to manually add the `sslutils` extension when using the following Postgres installations:

- PostgreSQL 10 or later (as distributed with the PEM server installer)
- Advanced Server 10 or later

Build `sslutils` with the same compiler that you used to compile the backend Postgres installation. If you're using a backend Postgres database that was installed on a Windows platform using a PostgreSQL one-click installer (from EnterpriseDB) or an EDB Postgres Advanced Server installer, you must use Visual Studio to build `sslutils`.

While specific details of the installation process vary by platform and compiler, the basic steps are the same. You must:

1. Copy the `sslutils` package to the Postgres installation directory.
2. Open the command line of the appropriate compiler, and navigate to the `sslutils` directory.
3. Use the following commands to build `sslutils` :

```
SET
USE_PGXS=1

SET GETTEXT_PATH=
<path_to_gettext>

SET OPENSSL_PATH=
<path_to_openssl>

SET PGPATH=
<path_to_pg_installation_dir>

SET ARCH=x86

REM Set ARCH x64 for 64
bit

msbuild sslutils.proj /p:Configuration=Release
```

Where:

`path_to_gettext` specifies the location of the `GETTEXT` library and header files.

`path_to_openssl` specifies the location of the `openssl` library and header files.

`path_to_pg_installation_dir` specifies the location of the Postgres installation.

4. Copy the compiled `sslutils` files to the appropriate directory for your installation. The `sslutils` directory contains the following files:

```
sslutils--1.3.sql

sslutils--unpackaged--1.3.sql

sslutils--pemagent.sql.in

sslutils.dll
```

Copy the `.dll` libraries and `.sql` files into place:

```
COPY sslutils*.sql*
"%PGPATH%\share\extension\"

COPY sslutils.dll "%PGPATH%\lib\"
```

## Registering the service

When you install a PostgreSQL or EDB Postgres Advanced Server database using an installer from EnterpriseDB, the installer registers the service for you.

If you're using Windows to host the PEM backend database, you must register the name of the Postgres server with the Windows service manager. If you're

using a Postgres server that was created using an EnterpriseDB installer, the service is registered automatically. If you are manually building the installation, you can use the `register` clause of the Postgres `pg_ctl` command to register the service:

```
pg_ctl register [-N <service_name>] [-U <user_name>]
| [-P <password>] [-D <data_directory>]
```

Where:

- `service name` specifies the name of the Postgres cluster.
- `user_name` specifies the name of an operating system user with sufficient privileges to access the Postgres installation directory and start the Postgres service.
- `password` specifies the operating system password associated with the user.
- `data_directory` specifies the location of the Postgres data directory.

For more information about using the `pg_ctl` command and the available command options, see the [Postgres core documentation](#).

## Invoking the PEM server installer

After preparing the existing Postgres server, invoke the PEM server installer. Assume administrator privileges and navigate to the directory that contains the installer. Then, invoke the installer:

```
pem_server-7.<x>.<x>-<x>-<platform>
```

Where `<x>` is the major and minor versions of PEM, and `<platform>` is the platform.

Then, begin the installation:

1. The installer displays a Welcome screen. Select **Next**.
2. Review the license agreement before selecting the appropriate radio button and accepting the agreement. Select **Next**.
3. Use the Installation Directory dialog box to specify the location of the PEM server and access the Advanced Options dialog box:
  - Use the **Installation Directory** field to open a browser dialog and select the directory to install the PEM server in.
  - If you're installing the PEM server on an existing server, select **Show advanced options** to include the Advanced Options dialog box in the installation process.
  - Select **Next**.
4. Use the Advanced Options dialog box to specify an installation type. Select:
  - **Web Services and Database** if both the Postgres server and the PEM-HTTPD server will reside on the current host. This option is valid if you're using an existing Postgres server to host the PEM server or using the PEM server installer to install the Postgres server where the PEM server will reside.  
  
If you select **Web Services and Database**, the PEM server installer checks the current host for a PEM-HTTPD installation and upgrades or installs PEM-HTTPD if necessary.
  - **Web Services** if only the PEM-HTTPD server will reside on the current host. See [Installing web services](#) for more information about invoking this option.
  - **Database** if you're installing only the PEM server (and creating the `pem` backend database) on the current host. This option is valid if you're using an existing Postgres server to host the PEM server or using the PEM server installer to install the PostgreSQL server where PEM will reside.



After selecting an installation option, select **Next**.

5. Use the list on the Database Server Selection dialog box to select a backend database for the PEM server:

- Select the name of a Postgres server on the current host that was installed using a Postgres one-click installer or EDB Postgres Advanced Server installer.
- Select **PostgreSQL x (Packaged)** to install and use the PostgreSQL server that's packaged with the PEM server installer, where **x** is the version of the PostgreSQL database server.
- Select **Other Database Server** to use a Postgres database that was installed from a source other than an EnterpriseDB installer (such as from an rpm or built from source).

#### Note

The selected database server must include an installation of the `sslutils` contrib module and a registered service (on Windows).

For more information, see [Preparing the Postgres server](#).

If you selected **Web Services and Database** on the Advanced Options dialog box, the installation wizard checks the current host for an existing PEM-HTTPD installation and upgrades or installs the service as needed.

If you selected **Database** on the Advanced Options dialog box, the Database Server Installation Details dialog box opens.

6. Use the fields on the Database Server Installation Details dialog box to describe the connection to the Postgres server that will host the PEM server:

- Enter the name of a database superuser in the **User** field.
- Enter the password associated with the superuser in the **Password** field.

Select **Next**.

7. Provide the administrator password for the PEM agent service to run under. Select **Next**.

8. Use the Network Details dialog box to specify the CIDR-style network address for the PEM agents to connect to the server (the `client-side` address). The specified address is added to the server's `pg_hba.conf` file.

You can specify more network addresses by manually adding entries to the `pg_hba.conf` file on the PostgreSQL server. Use the first entry as a template.

After you add the network address, select **Next**.

The PEM server installer installs a PEM agent to the host where the server resides to monitor the server and provide alert processing and garbage collection services. A certificate is also installed in the location specified in the **Agent certificate path** field.

9. You can enter an alternate description or an alternate agent certificate path for the PEM agent or accept the defaults. Select **Next**.

10. The wizard is now ready to install the PEM server. Select **Next** to continue with the installation.

During the installation process, the installer copies files to the system and sets up the PEM server's backend database. A confirmation indicates that the `pem` database was created and configured.

11. Select **OK**.

If you're using a PEM-HTTPD service that resides on a separate host, you must:

- Modify the `pg_hba.conf` file on the Postgres server host to allow connections between the hosts.
- Invoke the PEM server installer on the host of the PEM-HTTPD server. See [Installing Web Services](#) for more information about installing PEM-HTTPD.

## Invoking the server installer from the command line

The command line options of the PEM server and PEM agent installers offer functionality in situations where a graphical installation might not work because of limited resources or system configuration. You can:

- Include the `--mode unattended` option when invoking the installer to perform an installation without user input.

Not all command line options are suitable for all platforms. For a complete reference guide to the command line options, use the `--help` option when you invoke the installer.

## Invoking the PEM server installer in unattended mode

You can perform an unattended PEM server installation by providing installation preferences on the command line when invoking the installer. The system on which you're installing the PEM server must have internet access.

You must have administrator privileges to install the PEM server. Before invoking the PEM server installer, you must install the following dependencies:

- PostgreSQL
- pem-httpd
- Language Pack

You can use the PEM server installer to satisfy the dependencies of the PEM server. Navigate to the location of the installer, and use the following command to extract the dependencies:

```
pem-server-7.<x>.<x>-windows-x64.exe --extract-dependents C:\
```

In this example, the files are extracted to the `C:\` directory. After extracting the files, you must install each program. Navigate to the directory that contains the files (in this example, `C:\`), and enter:

```
edb-languagepack-<version>-windows-x64.exe --mode unattended
pem-httpd-<version>-windows-x64.exe --mode unattended
postgresql-<version>-windows-x64.exe --mode unattended
```

Then, you can invoke the PEM server installer:

```
pem-server-7.<x>.<x>-windows-x64.exe --mode unattended
--existing-user <registered_edb_user> --existing-password
<edb_user_password> --pgport <port> --pguser postgres
--agent_description pem-agent --systempassword <windows_password>
--agent-crt-path C:\edb``
```

Where:

- `registered_edb_user` specifies the name of a registered EnterpriseDB user. To register, visit the [EDB website](#).
- `edb_user_password` specifies the password associated with the EDB user account.
- `port` specifies the port used by the backing PostgreSQL database. By default, the PostgreSQL database uses port 5432.
- `cidr_address_range` specifies the address range to add to the `pg_hba.conf` file of the PEM server's backing database to allow connections from the agents for the server to monitor. You can specify a network range (for example, 192.168.2.0/24) to provide server access to agents that reside on the same network.
- `windows_password` specifies the password associated with the Windows administrator's account.

### Note

When invoked in unattended mode, the PostgreSQL installer creates a user named `postgres` with a password of `postgres`.

## 7.6 Creating an EDB repository on an isolated network

You can create a local repository to act as a host for the PEM RPM packages if the server where you want to upgrade PEM can't directly access the EDB repository. This is a high-level overview of the steps required. You might need to modify the process for your own network.

To create and use a local repository:

1. Use the following commands on a system with Internet access to download the dependencies for PEM:

```
yum install yum-plugin-downloadonly
mkdir /<pem_dir>
yum install --downloadonly --downloadaddir=/<pem_dir>/ edb-pem
mkdir /<epel_dir>
yum install --downloadonly --downloadaddir=/<epel_dir>/ epel-release*
```

Where `<pem_dir>` and `<epel_dir>` are the local directories that you create for downloading the RPMs.

2. Copy the directories `/<pem_dir>` and `/<epel_dir>` to the machine in the isolated network.
3. Create the repositories:

```
createrepo /<pem_dir>
createrepo /<epel_dir>
```

4. Create a repository configuration file called `/etc/yum.repos.d/pem.repo` with connection information that specifies:

```
[pemrepo]
name=PEM Repository
baseurl=file:///pem7/
enabled=1
gpgcheck=0
```

5. Create a repository configuration file called `/etc/yum.repos.d/epel.repo` with connection information that specifies:

```
[epelrepo]
name=epel Repository
baseurl=file:///pem7/
enabled=1
gpgcheck=0
```

6. After specifying the location and connection information for your local repository, you can use `dnf` commands to install or upgrade PEM server:

To install PEM server:

```
dnf install edb-pem
```

To upgrade PEM server:

```
dnf upgrade edb-pem
```

## 7.7 Configuring the PEM server on Linux

The PEM server package includes a script ( `configure-pem-server.sh` ) to help automate the configuration process for Linux platform installations. The script is installed in the `/usr/edb/pem/bin` directory. To invoke the script, use the command:

```
/usr/edb/pem/bin/configure-pem-server.sh
```

**Note**

If you're using the SSL certificates, then make sure that all the SSL certificates are in the data directory in the backend database server. If the certificates aren't in the data directory, then the PEM server's configure script might fail because it looks into the data directory while configuring the PEM server.

When invoking the script, you can include command line options to specify configuration properties. The script prompts you for values that you omit on the command line. The accepted options are:

Option	Description
<code>-acp</code>	Defines PEM agent certificate path. The default is <code>/root/.pem</code> .
<code>-ci</code>	CIDR-formatted network address range that agents connect to the server from, to be added to the server's <code>pg_hba.conf</code> file, for example, <code>192.168.1.0/24</code> . The default is <code>0.0.0.0/0</code> .
<code>-dbi</code>	The directory for the database server installation, for example, <code>/usr/edb/as12</code> for EDB Postgres Advanced Server or <code>/usr/pgsql-12</code> for PostgreSQL.
<code>-ds</code>	The unit file name of the PEM database server. For EDB Postgres Advanced Server, the default file name is <code>edb-as-12</code> . For PostgreSQL, it's <code>postgresql-12</code> .
<code>-ho</code>	The host address of the PEM database server.
<code>-p</code>	The port number of the PEM database server.
<code>-ps</code>	The service name of the pemagent. The default value is <code>pemagent</code> .
<code>-sp</code>	The superuser password of the PEM database server. This value is required.
<code>-su</code>	The superuser name of the PEM database server.
<code>-t</code>	The installation type: Specify <code>1</code> if the configuration is for web services and backend database, <code>2</code> if you're configuring web services, or <code>3</code> if you're configuring the backend database. If you specify <code>3</code> , the database must reside on the local host.

If you don't provide configuration properties on the command line, the script prompts you for values. When you invoke the script, choose from:

- `Web Services and Database` — Select this option if the web server and database both reside on the same host as the PEM server.
- `Web Services` — Select this option if the web server resides on a different host from the PEM server.
- `Database` — Select this option to configure the PEM backend database for use by the PEM server. The specified database must reside on the local host.

**Note**

If the web server and the backend database (PEM server) reside on separate hosts, configure the database server first (option 3) and then web services (option 2). The script proceeds only if the backend database is configured before web services.

After selecting a configuration option, the script prompts you for configuration properties. When the script completes, it creates the objects required by the PEM server or performs the configuration steps required. To view help for the script, use the command:

```
/usr/edb/pem/bin/configure-pem-server.sh -help
```

## Post-configuration steps when web server and PEM backend database are installed separately

If you choose to run the web application server on a separate host from the backend database, you need to perform some additional manual steps before PEM is fully operational.

Make sure that the backend Postgres database accepts the connections from any user permitted to log in to PEM from the web application server. To achieve this, add this entry to `pg_hba.conf`:

```
host pem +pem_user <web_app_ip>/32
md5
```

Where `<web_app_ip>` is the IP address of the web application server.

Additionally, if the IP address of the web application server isn't within the network address range specified when the script is executed, you must add two entries to allow the PEM agent on this server to connect:

```
host pem +pem_agent <web_app_ip>/32
md5
host pem +pem_agent <web_app_ip>/32
cert
```

Where `<web_app_ip>` is the IP address of the web application server.

## Accessing the PEM application

After configuring the PEM server, you can access the PEM web interface in your browser. Navigate to:

```
https://<ip_address_of_PEM_server>:8443/pem
```

By default, the web services listen on port 8443. To change the port, see [Changing the default port](#).

## 8 Installing Postgres Enterprise Manager agent

Select a link to access the applicable installation instructions:

### Linux x86-64 (amd64)

#### Red Hat Enterprise Linux (RHEL) and derivatives

- [RHEL 9, RHEL 8](#)
- [Oracle Linux \(OL\) 9, Oracle Linux \(OL\) 8](#)

- [Rocky Linux 9, Rocky Linux 8](#)
- [AlmaLinux 9, AlmaLinux 8](#)

#### SUSE Linux Enterprise (SLES)

- [SLES 15, SLES 12](#)

#### Debian and derivatives

- [Ubuntu 22.04, Ubuntu 20.04](#)
- [Debian 11](#)

#### Linux [IBM Power \(ppc64le\)](#)

#### Red Hat Enterprise Linux (RHEL) and derivatives

- [RHEL 9, RHEL 8](#)

#### SUSE Linux Enterprise (SLES)

- [SLES 15, SLES 12](#)

#### Windows

- [Windows Server 2019](#)

## 8.1 Installing Postgres Enterprise Manager agent on Linux x86 (amd64)

Operating system-specific install instructions are described in the corresponding documentation:

#### Red Hat Enterprise Linux (RHEL) and derivatives

- [RHEL 9](#)
- [RHEL 8](#)
- [Oracle Linux \(OL\) 9](#)
- [Oracle Linux \(OL\) 8](#)

- [Rocky Linux 9](#)
- [Rocky Linux 8](#)
- [AlmaLinux 9](#)
- [AlmaLinux 8](#)

#### SUSE Linux Enterprise (SLES)

- [SLES 15](#)
- [SLES 12](#)

#### Debian and derivatives

- [Ubuntu 22.04](#)
- [Ubuntu 20.04](#)
- [Debian 11](#)

### 8.1.1 Installing Postgres Enterprise Manager agent on RHEL 9 or OL 9 x86\_64

#### Prerequisites

Before you begin the installation process:

- Set up the EDB repository.

Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

To determine if your repository exists, enter this command:

```
dnf repolist | grep enterprisedb
```

If no output is generated, the repository isn't installed.

To set up the EDB repository:

1. Go to [EDB repositories](#).
2. Select the button that provides access to the EDB repository.
3. Select the platform and software that you want to download.
4. Follow the instructions for setting up the EDB repository.

## Install the package

```
sudo dnf -y install edb-pem-agent
```

After installing PEM agent, you need to register the PEM agent. For detailed information see [Registering an agent](#).

## 8.1.2 Installing Postgres Enterprise Manager agent on RHEL 8 or OL 8 x86\_64

### Prerequisites

Before you begin the installation process:

- Set up the EDB repository.

Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

To determine if your repository exists, enter this command:

```
dnf repolist | grep enterprisedb
```

If no output is generated, the repository isn't installed.

To set up the EDB repository:

1. Go to [EDB repositories](#).
2. Select the button that provides access to the EDB repository.
3. Select the platform and software that you want to download.
4. Follow the instructions for setting up the EDB repository.

## Install the package

```
sudo dnf -y install edb-pem-agent
```

After installing PEM agent, you need to register the PEM agent. For detailed information see [Registering an agent](#).

## 8.1.3 Installing Postgres Enterprise Manager agent on AlmaLinux 9 or Rocky

### Prerequisites

Before you begin the installation process:



- Set up the EDB repository.

Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

To determine if your repository exists, enter this command:

```
dnf repolist | grep enterprisedb
```

If no output is generated, the repository isn't installed.

To set up the EDB repository:

1. Go to [EDB repositories](#).
2. Select the button that provides access to the EDB repository.
3. Select the platform and software that you want to download.
4. Follow the instructions for setting up the EDB repository.

## Install the package

```
sudo dnf -y install edb-pem-agent
```

After installing PEM agent, you need to register the PEM agent. For detailed information see [Registering an agent](#).

### 8.1.4 Installing Postgres Enterprise Manager agent on AlmaLinux 8 or Rocky

#### Prerequisites

Before you begin the installation process:

- Set up the EDB repository.

Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

To determine if your repository exists, enter this command:

```
dnf repolist | grep enterprisedb
```

If no output is generated, the repository isn't installed.

To set up the EDB repository:

1. Go to [EDB repositories](#).
2. Select the button that provides access to the EDB repository.

3. Select the platform and software that you want to download.
4. Follow the instructions for setting up the EDB repository.

## Install the package

```
sudo dnf -y install edb-pem-agent
```

After installing PEM agent, you need to register the PEM agent. For detailed information see [Registering an agent](#).

## 8.1.5 Installing Postgres Enterprise Manager agent on SLES 15 x86\_64

### Note

Postgres Enterprise Manager 8.3 and later is supported on SLES.

## Prerequisites

Before you begin the installation process:

- Set up the EDB repository.

Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

To determine if your repository exists, enter this command:

```
zypper lr -E | grep enterprisedb
```

If no output is generated, the repository isn't installed.

To set up the EDB repository:

1. Go to [EDB repositories](#).
2. Select the button that provides access to the EDB repository.
3. Select the platform and software that you want to download.
4. Follow the instructions for setting up the EDB repository.

- Activate the required SUSE module:

```
# You can use SLES 15 SP3 for PEM 8.3 and later:
sudo SUSEConnect -p PackageHub/15.3/x86_64

# You can use SLES 15 SP4 for PEM 8.6 and later:
sudo SUSEConnect -p PackageHub/15.4/x86_64
```

- Refresh the metadata:

```
sudo zypper refresh
```

## Install the package

```
sudo zypper -n install edb-pem-agent
```

After installing PEM agent, you need to register the PEM agent. For detailed information see [Registering an agent](#).

## 8.1.6 Installing Postgres Enterprise Manager agent on SLES 12 x86\_64

### Note

Postgres Enterprise Manager 8.3 and later is supported on SLES.

## Prerequisites

Before you begin the installation process:

- Set up the EDB repository.

Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

To determine if your repository exists, enter this command:

```
zypper lr -E | grep enterprisedb
```

If no output is generated, the repository isn't installed.

To set up the EDB repository:

1. Go to [EDB repositories](#).
2. Select the button that provides access to the EDB repository.
3. Select the platform and software that you want to download.
4. Follow the instructions for setting up the EDB repository.

- Activate the required SUSE module:

```
sudo SUSEConnect -p PackageHub/12.5/x86_64  
sudo SUSEConnect -p sle-sdk/12.5/x86_64
```

- Refresh the metadata:

```
sudo zypper refresh
```

## Install the package

```
sudo zypper -n install edb-pem-agent
```

After installing PEM agent, you need to register the PEM agent. For detailed information see [Registering an agent](#).

## 8.1.7 Installing Postgres Enterprise Manager agent on Ubuntu 22.04 x86\_64

### Prerequisites

Before you begin the installation process:

- Set up the EDB repository.

Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

To determine if your repository exists, enter this command:

```
apt-cache search enterprisedb
```

If no output is generated, the repository isn't installed.

To set up the EDB repository:

1. Go to [EDB repositories](#).
2. Select the button that provides access to the EDB repository.
3. Select the platform and software that you want to download.
4. Follow the instructions for setting up the EDB repository.

## Install the package

```
sudo apt-get -y install edb-pem-agent
```

After installing PEM agent, you need to register the PEM agent. For detailed information see [Registering an agent](#).

## 8.1.8 Installing Postgres Enterprise Manager agent on Ubuntu 20.04 x86\_64

## Prerequisites

Before you begin the installation process:

- Set up the EDB repository.

Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

To determine if your repository exists, enter this command:

```
apt-cache search enterprisedb
```

If no output is generated, the repository isn't installed.

To set up the EDB repository:

1. Go to [EDB repositories](#).
2. Select the button that provides access to the EDB repository.
3. Select the platform and software that you want to download.
4. Follow the instructions for setting up the EDB repository.

## Install the package

```
sudo apt-get -y install edb-pem-agent
```

After installing PEM agent, you need to register the PEM agent. For detailed information see [Registering an agent](#).

## 8.1.9 Installing Postgres Enterprise Manager agent on Debian 11 x86\_64

### Prerequisites

Before you begin the installation process:

- Set up the EDB repository.

Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

To determine if your repository exists, enter this command:

```
apt-cache search enterprisedb
```

If no output is generated, the repository isn't installed.

To set up the EDB repository:

1. Go to [EDB repositories](#).
2. Select the button that provides access to the EDB repository.
3. Select the platform and software that you want to download.
4. Follow the instructions for setting up the EDB repository.

### Install the package

```
sudo apt-get -y install edb-pem-agent
```

After installing PEM agent, you need to register the PEM agent. For detailed information see [Registering an agent](#).

## 8.2 Installing Postgres Enterprise Manager agent on Linux IBM Power (ppc64le)

Operating system-specific install instructions are described in the corresponding documentation:

### Red Hat Enterprise Linux (RHEL)

- [RHEL 9](#)
- [RHEL 8](#)

### SUSE Linux Enterprise (SLES)

- [SLES 15](#)
- [SLES 12](#)

### 8.2.1 Installing Postgres Enterprise Manager agent on RHEL 9 ppc64le

#### Prerequisites

Before you begin the installation process:

- Set up the EDB repository.

Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

To determine if your repository exists, enter this command:

```
dnf repolist | grep enterprisedb
```

If no output is generated, the repository isn't installed.

To set up the EDB repository:

1. Go to [EDB repositories](#).
2. Select the button that provides access to the EDB repository.
3. Select the platform and software that you want to download.
4. Follow the instructions for setting up the EDB repository.

## Install the package

```
sudo dnf -y install edb-pem-agent
```

After installing PEM agent, you need to register the PEM agent. For detailed information see [Registering an agent](#).

## 8.2.2 Installing Postgres Enterprise Manager agent on RHEL 8 ppc64le

### Prerequisites

Before you begin the installation process:

- Set up the EDB repository.

Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

To determine if your repository exists, enter this command:

```
dnf repolist | grep enterprisedb
```

If no output is generated, the repository isn't installed.

To set up the EDB repository:

1. Go to [EDB repositories](#).
2. Select the button that provides access to the EDB repository.
3. Select the platform and software that you want to download.
4. Follow the instructions for setting up the EDB repository.

## Install the package

```
sudo dnf -y install edb-pem-agent
```

After installing PEM agent, you need to register the PEM agent. For detailed information see [Registering an agent](#).

### 8.2.3 Installing Postgres Enterprise Manager agent on SLES 15 ppc64le

#### Note

Postgres Enterprise Manager 8.3 and later is supported on SLES.

## Prerequisites

Before you begin the installation process:

- Set up the EDB repository.

Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

To determine if your repository exists, enter this command:

```
zypper lr -E | grep enterprisedb
```

If no output is generated, the repository isn't installed.

To set up the EDB repository:

1. Go to [EDB repositories](#).
2. Select the button that provides access to the EDB repository.
3. Select the platform and software that you want to download.
4. Follow the instructions for setting up the EDB repository.

- Activate the required SUSE module:

```
# You can use SLES 15 SP3 for PEM 8.3 and later:  
sudo SUSEConnect -p PackageHub/15.3/ppc64le  
  
# You can use SLES 15 SP4 for PEM 8.6 and later:  
sudo SUSEConnect -p PackageHub/15.4/ppc64le
```

- Refresh the metadata:

```
sudo zypper refresh
```



## Install the package

```
sudo zypper -n install edb-pem-agent
```

After installing PEM agent, you need to register the PEM agent. For detailed information see [Registering an agent](#).

## 8.2.4 Installing Postgres Enterprise Manager agent on SLES 12 ppc64le

### Note

Postgres Enterprise Manager 8.3 and later is supported on SLES.

## Prerequisites

Before you begin the installation process:

- Set up the EDB repository.

Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

To determine if your repository exists, enter this command:

```
zypper lr -E | grep enterprisedb
```

If no output is generated, the repository isn't installed.

To set up the EDB repository:

1. Go to [EDB repositories](#).
2. Select the button that provides access to the EDB repository.
3. Select the platform and software that you want to download.
4. Follow the instructions for setting up the EDB repository.

- Activate the required SUSE module:

```
sudo SUSEConnect -p PackageHub/12.5/ppc64le  
sudo SUSEConnect -p sle-sdk/12.5/ppc64le
```

- Refresh the metadata:

```
sudo zypper refresh
```

## Install the package

```
sudo zypper -n install edb-pem-agent
```

After installing PEM agent, you need to register the PEM agent. For detailed information see [Registering an agent](#).

## 8.3 Installing a PEM agent on Windows

The PEM agent graphical installer for Windows installs and registers the PEM agent.

To invoke the PEM agent installer, assume administrator privileges and navigate to the directory that contains the installer. Then, invoke the installer:

```
pem_agent-8.<x>.<x>-<x>-platform.exe
```

1. The Setup screen opens, welcoming you to the PEM agent installer. Select **Next**.
2. Review the license agreement before selecting the appropriate radio button and accepting the agreement. Select **Next**.
3. By default, the PEM agent is installed in the `C:\Program Files (x86)\edb\pem` directory. You can accept the default installation directory. Or you can modify the contents of the **Installation Directory** field, specifying an alternate installation directory for the PEM agent.
  - By default, the PEM agent installer places a certificate in the Administrator's `%APPDATA%\pem` directory. Select **Show advanced options** to include a dialog box in the installation process that allows you to specify an alternate path for the certificate file.
  - Select **Register now?** to register the newly installed PEM agent with the PEM server.
  - Select **Next**.
4. Enter the connection details for the PEM server in the PEM Server Installation Details dialog box.
  - Specify the name or IP address of the system where the PEM database server resides in the **Host** field. If the PEM-HTTPD web server and PEM database are hosted on different systems, you must specify the host of the PEM database.
  - Specify the name of the database superuser in the **User Name** field.
  - Specify the password associated with the database superuser in the **Password** field.
  - Specify the port that PostgreSQL is monitoring in the **Port** field.
5. Select **Next** to continue to pemAgent Service Account. The installer attempts to connect to the server to verify that the details are correct.

### Note

The PEM server must allow connections from the PEM agent installer. If you encounter a connection error, confirm the connection properties specified in the PEM Server Installation Details dialog box are correct. Confirm that the `pg_hba.conf` file on the PEM server allows a connection to the server described in the error message.

6. Provide the password for the edb account for the pemAgent service to run under. The agent certificate and key files are created in the `C:\Users\edb\AppData\Roaming\pem` directory. Select **Next**.
7. The tree displayed in the Browser panel of the PEM web interface displays the value entered in the **Description** field to identify the PEM agent. Specify a descriptive name for the agent, such as the hostname of the machine the agent is installed on or a name that reflects the host's functionality.

Provide a descriptive name, or accept the default provided by the PEM Agent host. Select **Next**.

8. If you selected **Show advanced options**, the Advanced Options dialog box opens. By default, the PEM agent installer places the certificate in the `C:\Program Files (x86)\edb\pem` directory. Specify an alternate path for the certificate, or accept the default and select **Next**.

9. The wizard is ready to install the PEM agent. Select **Next**.
10. On the Ready to Install dialog box, select **Next**. The installer copies files to the system and registers the agent on the PEM server.

When the installation completes, the PEM agent is running and reporting operating system and host data to the PEM server. To start monitoring Postgres instances on the host of the PEM agent, add them to PEM's enterprise directory and bound them to the agent.

## Invoking the agent installer from the command line

The command line options of the PEM Agent installers offer functionality in situations where a graphical installation might not work because of limited resources or system configuration. You can:

- Include the `--mode unattended` option when invoking the installer to perform an installation without user input.

Not all command line options are suitable for all platforms. For a complete reference guide to the command line options, include the `--help` option when you invoke the installer.

## Invoking the PEM agent installer in unattended mode

You can perform an unattended PEM server installation by providing installation preferences on the command line when invoking the installer. The system where you're installing the PEM server must have internet access.

Before invoking the PEM agent installer in unattended mode, you must:

- Install the PEM server. The `pg_hba.conf` file of the PEM server must allow connections from the host of the PEM agent.
- Ensure that the monitored Postgres database has SSL enabled and is accepting connections.

You need administrator privileges to install the PEM Agent. Use the following command to invoke the PEM agent installer in unattended mode:

```
pem-agent-8.<x>.<x>-windows-x64.exe --mode unattended
--pghost <pem_server_host_address> --pgport <pem_server_port>
--pguser postgres --pgpassword <pguser_password>
--agent_description <agent_name>
```

Where:

- `pem_server_host_address` specifies the IP address of the host of the PEM server.
- `pem_server_port` specifies the port used by the backing PEM database. By default, the database uses port 5432.
- `pguser_password` specifies the password associated with the PEM database superuser.
- `agent_name` specifies a descriptive name for the PEM agent.

### Note

When configuring a shell/batch script run by a Windows agent, set the `AllowBatchJobSteps` parameter to `True` in the `agent.cfg` file. The PEM agent doesn't execute any batch/shell script by default.

## 9 Upgrade and migration

You can upgrade the Postgres Enterprise Manager (PEM) components including the PEM installation, the backend database, and the SQL Profiler. You can also move the PEM server.

**Note**

If you're upgrading PEM 7.x to the latest version and the PEM backend database is PostgreSQL or EDB Postgres Advanced Server version 10 or earlier:

1. Upgrade the PEM backend database.
2. Upgrade the PEM installation.

If the PEM backend database is PostgreSQL or EDB Postgres Advanced Server version 11 or later, then upgrading the PEM backend database is optional.

## 9.1 Upgrading a PEM installation

The process of upgrading a PEM installation is platform specific. You can update a PEM agent or server on a Linux platform by using the native packages and on a Windows platform by using the PEM graphical installer available for Windows.

**Note**

- Prior to PEM 7.8 release, you installed PEM agent or server on Linux either by using the graphical installer or by using the RPMs.
- For PEM version 7.8 and later, PEM graphical installers for Linux are discontinued. To upgrade a PEM agent or server on a Linux host from any earlier version to PEM version 7.9 or later, you must use native packages.

Links to PEM installers and RPMs are available at the [EDB website](#).

### 9.1.1 Upgrading a PEM native package installation on a Linux host

To upgrade PEM component software on Linux hosts, install a newer version of the PEM component native packages in the following order:

1. Invoke the PEM agent native package installer on each monitored node except the PEM server host.
2. Invoke the PEM server native package installer. It upgrades both the PEM server and the PEM agent that resides on the PEM server host.

During an installation, the component installation automatically detects an existing installation and performs an upgrade. After upgrading the PEM agent and server, you can upgrade SQL Profiler if required. That step is platform specific.

**Note**

If you already configured or are planning to configure any shell/batch script run by a Linux agent that's upgraded from any earlier version to version 7.11 or later, you must specify the user for the `batch_script_user` parameter in the `agent.cfg` file. We strongly recommend that you use a non-root user to run the scripts. Using the root user can result in compromising the data security and operating system security. However, if you want to restore the pemagent to its original settings using a root user to run the scripts, then you must set the `batch_script_user` parameter to `root`.

#### Prerequisites to upgrade a PEM installation on Linux host

PEM depends on third-party components from the vendor repository, including python3, libboost, openssl, snmp++, and libcurl. To ensure these components are up to date, update your operating system using platform-specific commands.

The minimum version required for openssl is 1.0.2k. If you're using a version of PostgreSQL or EDB Postgres Advanced Server earlier than version 10, before the upgrade you must install the `libs` package for version 10 or above on the system where the PEM server is installed. Use the following platform-specific commands to install the `libs` version 10 or above on your host.

### Prerequisites to upgrade a PEM installation on a RHEL host

To upgrade packages on a Rocky Linux or AlmaLinux or RHEL 8.x host:

```
dnf update
dnf upgrade
```

To upgrade EDB Postgres Advanced Server libs:

```
yum install edb-as<X>-server-libs
```

To upgrade PostgreSQL libs:

```
yum install postgresql<X>-libs
```

Where `<X>` is the PostgreSQL or EDB Postgres Advanced Server version whose `libs` package you want to install.

### Prerequisites to upgrade a PEM installation on a Debian or Ubuntu host

To upgrade packages on a Debian or Ubuntu host:

```
apt-get update
apt-get upgrade
```

To upgrade Advanced Server libs:

```
apt-get install edb-as<X>-server-libs
```

To upgrade PostgreSQL libs:

```
apt-get install postgresql<X>-libs
```

Where `<X>` is the PostgreSQL or EDB Postgres Advanced Server version whose `libs` package you want to install.

### Prerequisites to upgrade a PEM installation on a SLES host

To upgrade packages on a SLES host:

```
zypper update
zypper upgrade
```

To upgrade EDB Postgres Advanced Server libs:

```
zypper install edb-as<x>-server-libs
```

To upgrade PostgreSQL libs:

```
zypper install postgresql<x>-libs
```

Where `<X>` is the PostgreSQL or EDB Postgres Advanced Server version whose `libs` package you want to install.

## Upgrading a PEM agent native package installation

You can use native packages to upgrade existing PEM agents initially installed using native packages. The upgrade process doesn't update the PEM agent configuration file. After installing the new agent, you must manually copy the configuration file of the existing agent to the new installation location.

### Upgrading the PEM agent on a RHEL host

For RHEL 8.x, to upgrade a PEM agent, use the following command:

```
dnf upgrade edb-pem-agent
```

### Upgrading a PEM agent on a Debian or Ubuntu host

To upgrade a PEM agent, use the following command:

```
apt-get upgrade edb-pem-agent
```

### Upgrading a PEM agent on a SLES host

To upgrade a PEM agent, use the following command:

```
zypper update edb-pem-agent
```

## Upgrading a PEM server native package installation

If you initially used native packages to install your PEM server, you can use native packages to upgrade your PEM server. The commands to upgrade are platform specific.

If you want to upgrade a PEM server that is installed on a machine in an isolated network, you need to create a PEM repository on that machine before you upgrade the PEM server. For more information about creating a PEM repository on an isolated network, see [Creating a PEM Repository on an Isolated Network](#).

### Upgrading a PEM server on a Rocky Linux, AlmaLinux, or RHEL host

To use an RPM package to upgrade an existing RPM installation, you can use the `dnf` package manager to upgrade the installed version of the PEM server on Rocky Linux/AlmaLinux/RHEL 8.x:

```
dnf upgrade edb-pem
```

After upgrading the PEM server using the `dnf` command, you must configure the PEM server. For detailed information, see [Configuring the PEM server](#).

### Upgrading the PEM server on a Debian or Ubuntu host

You can use the `apt-get` package manager to upgrade the installed version of the PEM server on supported versions of Debian or Ubuntu:

```
apt-get upgrade edb-pem
```

After upgrading the PEM server with `apt-get`, you need to configure the PEM server. For detailed information, see [Configuring the PEM server](#).

### Upgrading the PEM server on a SLES host

You can use the zypper package manager to upgrade the installed version of the PEM Server on supported versions of a SLES host:

```
zypper update edb-pem
```

After upgrading the PEM server using zypper, you need to configure the PEM server. For detailed information, see [Configuring the PEM server](#).

#### Note

If you upgrade the PEM backend database server and the PEM server, update the `PG_INSTALL_PATH` and `DB_UNIT_FILE` parameters pointing to the new version in the `/usr/edb/pem/share/.install-config` file before you run the configure script.

## Configuring the PEM server

After upgrading the PEM server, you can use the following command to configure the PEM server:

```
/usr/edb/pem/bin/configure-pem-server.sh
```

The configure script uses the values from the old PEM server configuration file while running the script.

After executing the PEM server configuration file, use your version-specific service control command to restart the httpd service.

For detailed information, see [Configuring the PEM server on Linux platforms](#).

#### Note

From PEM version 7.11 and later, the configure script requires a superuser password only after the upgrade process.

## 9.1.2 Upgrading a PEM installation on a Windows host

To upgrade PEM component software on Windows hosts, invoke a newer version of the PEM component installers in the following order:

1. Invoke the PEM agent installer on each monitored node except the PEM server host.
2. Invoke the PEM server installer. This installer upgrades both the PEM server and the PEM agent that resides on the PEM server host.

During an installation, the component installer automatically detects an existing installation and performs an upgrade. After upgrading the PEM agent and server, you can upgrade SQL profiler if required. This step is platform specific.

## Upgrading a PEM agent on a Windows host

To upgrade a system that is currently monitored by a PEM agent to a more recent PEM agent, download and invoke a newer version of the PEM Agent installer on the system that the agent is monitoring.

1. To invoke the installer, right-click the downloaded installer icon and select **Run as Administrator**. The PEM Agent Setup wizard opens, welcoming you.
2. Read and accept the license agreement and then select **Next**.
3. The setup wizard automatically detects an existing agent and upgrades the installed version. Select **Next**.
4. The pemAgent Service Account dialog box might prompt you for the password of the account under which the PEM agent service runs. If prompted, provide the password, and select **Next**.
5. When the Ready to Install dialog box informs you that the installation is about to begin, select **Next**. The wizard upgrades your PEM agent to the latest version.
6. The PEM Agent Setup wizard informs you when the installation completes. Select **Finish**.
7. After the installation completes, you're prompted to restart the machine. Select **Yes** to restart the machine and the PEM agent.

## Upgrading the PEM server on a Windows host

The PEM server installer enables you to upgrade between major versions of the PEM server. You can upgrade from version 5.0 to version 7.16 without first upgrading to version 6.0.

1. To invoke the installer, right-click the downloaded installer, and select **Run as Administrator**.
2. The PEM Server Setup wizard welcomes you. Select **Next**.
3. The PEM Server Setup wizard prompts you to accept the license agreement. After reviewing the license agreement, select **I accept the agreement** and select **Next**.
4. The wizard checks the PEM server host for an existing PEM server installation. If the wizard locates an installation, it performs an upgrade. Select **Next**.

Before upgrading the PEM server, the wizard confirms that the requirements of the new PEM server are present. If any supporting components are missing or are at a version that doesn't support the new PEM installation, the wizard informs you that it must upgrade the dependencies. It then invokes the required installers.

5. When the installation wizards completes the dependency upgrades, you're prompted to restart the machine. Select **No** to continue the upgrade process.
6. The wizard then opens the Database Server Installation Details dialog box, prompting you for connection credentials for the database superuser of the PEM backend database. Provide:
  - The name of the database superuser in the **User** field.
  - The password associated with the database superuser in the **Password** field.

Select **Next**.

1. The pemAgent Service Account dialog box might prompt you for the password of the account under which the PEM agent service runs. If prompted,



provide the password, and select **Next**.

2. The Ready to Install dialog box informs you that the setup wizard is ready to perform the installation. Select **Next** to start the installation.

After upgrading the PEM server (and the agent that resides on the same host as the PEM server) and configuring the web service, the PEM setup wizard notifies you of the port on which the service is listening. Use this port number when connecting to the PEM server with the PEM client.

3. Select **OK**. The PEM server setup wizard informs you that the installation is complete.
4. If you're prompted to restart the machine, select **Yes** to restart the machine and the httpd service.

If you installed the PEM backend database server and PEM-HTTPD on different hosts, then you must run the PEM server installer twice: once on each host. Extract the language pack installer, and install it on the host of PEM-HTTPD before invoking the PEM installer. Include the following keywords when invoking the installer to extract the language pack:

```
--extract-languagepack <path>
```

Where `<path>` specifies an existing path for extracting the language pack installer.

#### Note

By default EDB Language Pack is installed in `C:\edb\languagepack\v1`.

If you're upgrading the PEM Server using StackBuilder Plus, then you might see an error. After displaying the error, PEM reports that installation is completed. However, the installation isn't complete. You need to do the installation by invoking the installer file from the location where it is downloaded.

After upgrading the PEM server, you might want to upgrade the backend database to a more recent version. For information, see [Upgrading the backend Postgres database](#).

## 9.2 Upgrading the PEM backend Postgres database

If you're updating PEM components and the PEM backend database, perform PEM component updates on the server and agent before updating the backend database. For more information about updating PEM component software, see [Upgrading a PEM installation](#).

#### Note

- From PEM 8.0 onwards, PostgreSQL or EPAS versions 11 or later are only supported as backend database servers. As a result, if your backend database server is earlier than version 11, you need to first upgrade your backend database server and then upgrade the PEM components.
- After upgrading the backend database server, if you encounter this error while creating the server in the PEM web interface:

```
Error - User does not have enough permission to add new server.
Please contact the administrator to grant 'pem_database_server_registration' role
to the 'enterprisedb' user.
```

Resolve the error by updating the roles and granting appropriate permissions:

```
UPDATE pem.roles SET rolid = pr.oid FROM pg_roles pr WHERE pr.rolname = 'pem_' ||
component;
```

The update process uses the pg\_upgrade utility to migrate from one version of the backend server to a more recent version. pg\_upgrade enables migration between any supported version of Postgres and any subsequent release of Postgres that's supported on the same platform.

If the source PEM server is earlier than the 7.16 version, then you need to replace the following functions before you run pg\_upgrade:

- The `abstime`, `reltime`, and `tinterval` datatypes are deprecated from Postgres version 12 or later, hence to replace those datatypes with `timestampz` data type use this command:

```
DO
$$
DECLARE
    rec
record;
    cnt
integer;
BEGIN
    -- Check for the deprecated type in our user info
    probe
    SELECT count(*) INTO
    cnt
    FROM
    pem.probe_column
    WHERE sql_data_type = 'abstime' AND internal_name =
    'valuntil';
    IF cnt = 0
    THEN
        RETURN;
    END IF;
    ALTER TABLE pemdata.user_info
    ALTER COLUMN valuntil SET DATA TYPE
timestampz;
    ALTER TABLE pemhistory.user_info
    ALTER COLUMN valuntil SET DATA TYPE
timestampz;
    -- Now update the pem.probe_column
    itself
    UPDATE
    pem.probe_column
    SET sql_data_type = 'timestampz'
    WHERE sql_data_type = 'abstime' AND internal_name =
    'valuntil';
END;
$$ LANGUAGE
'plpgsql';
```

- Replace this function to avoid any alert errors:

```
CREATE OR REPLACE FUNCTION
pem.check_alert_params_array_size(
template_id pem.alert_template.id%type, params
text[]
)
RETURNS bool AS
$FUNC$
DECLARE
    res bool :=
TRUE;
BEGIN
    /*
    * During restoring the pem database, it does not maintain the order
    while
    * inserting data in the table, and uses the sort table based on
    the
    *
    names.
    * Hence - we need to check the foreign key constraint is present
    before
    * validating these
    values.
    */
END;
```

```

    */
    IF EXISTS(
        SELECT 1 FROM information_schema.table_constraints
        WHERE constraint_name='alert_template_id_fkey' AND
        table_name='alert' AND table_schema='pem'
    )
    THEN
        /*
        * Need to use the IS TRUE construct outside the main query,
        because
        * otherwise if there's no template by that ID then the query would
        return
        * 0 rows and the result of the function would be undefined and
        CHECK
        * constraint would
        succeed.
        * Probably this is being over-cautious, because
        pem.alert.template_id
        * references pem.alert_template.id. But the SQL standard (probably)
        does
        * not define the order in which the CHECK or the FOREIGN KEY
        constraints
        * should be validated; in case CHECK is validated first, we want it
        to
        *
        fail.
        */
    EXECUTE $$SQL$
        SELECT
        (
            SELECT pem.check_array_size_equal(t.param_names,
$2)
            FROM pem.alert_template AS
            t
            WHERE id =
$1
        ) IS
    TRUE
    $$SQL$ INTO res USING template_id,
    params;
END IF;
    RETURN res;
END
$FUNC$ LANGUAGE
'plpgsql';

```

pg\_upgrade supports a transfer of data between servers of the same type. For example, you can use pg\_upgrade to move data from a PostgreSQL 10 backend database to a PostgreSQL 11 backend database but not to an EDB Postgres Advanced Server 11 backend database. If you want to migrate to a different type of backend database (such as from a PostgreSQL server to EDB Postgres Advanced Server), see [Moving the Postgres Enterprise Manager server](#).

You can find more information about using pg\_upgrade at [pg\\_upgrade](#).

1. Download and invoke the updated installer. Installers for PostgreSQL and EDB Postgres Advanced Server are available through the [EDB website](#).

After downloading the installer for the server version you are upgrading to, invoke the installer on the host of the PEM server. Follow the onscreen instructions of the installation wizard to configure and install the Postgres server.

You can optionally use a custom-built PostgreSQL server as a host of the PEM backend database. If you're upgrading from a PostgreSQL backend database listening on port 5432, the new server must be configured to listen on a different port.

2. Configure SSL utilities on the new server. The new backend database must be running the same version of sslutils that the current backend database is running. You can download the SSL Utils package from the [EDB website](#).

You don't need to manually add the sslutils extension when using the EDB Postgres Advanced Server as the new backend database. The process of configuring sslutils is platform specific.

## On Linux

- On an EDB Postgres Advanced Server backend database, the sslutils extension is installed by default.
- If you're using PostgreSQL as a PEM backend database, verify that you have access to the PostgreSQL community repository, and use the command:

```
yum install sslutils_<X>
```

Where `<X>` is the server version.

- If you're using a EDB one-click installer of PostgreSQL as a PEM backend database, use the command:

```
yum install gcc openssl-devel
```

Set the value of PATH so it can locate the pg\_config program

```
export
PATH=$PATH:/opt/postgres_inst_dir/<X>/bin/
```

Move into the `sslutils` folder, and enter:

```
make
USE_PGXS=1
make USE_PGXS=1
install
```

Use psql to create the sslutils extension

```
CREATE EXTENSION
sslutils
```

Debian 10 and Ubuntu 20 have increased the requirements to accept the certificates for security reasons. If you want to install the PEM agent on any of the machines, you must upgrade sslutils to 1.3 where 4096-bit RSA key and sha256 signature algorithm support was added. If you don't upgrade sslutils to 1.3, then PEM agent might fail to connect to the PEM backend database server, and it might log the error "ca md too weak."

## On Windows

- You must compile sslutils on the new backend database with the same compiler that was used to compile sslutils on the original backend database. If you're moving to a Postgres database that was installed using a PostgreSQL one-click installer (from EDB) or an EDB Postgres Advanced Server installer, use Visual Studio to build sslutils. If you're upgrading to PostgreSQL 10 or later, use Visual Studio 2010.

For detailed information about building a specific version of Postgres on Windows, consult the core documentation for that version. Core documentation is available at the [PostgreSQL project website](#).

- While specific details of the process vary by platform and compiler, the basic steps on each platform are the same. The example that follows shows compiling OpenSSL support for PostgreSQL on a 32-bit Windows system.

Before compiling the OpenSSL extension, you must locate and install OpenSSL for your version of Windows. Before invoking the OpenSSL installer you might need to download and install a prerequisite redistributable (such as `vcredist_x86.exe`).

After installing OpenSSL, download and unpack the [sslutils utility package](#).

- Copy the unpacked `sslutils` folder to the Postgres installation directory (`C:\ProgramFiles\PostgreSQL\<x.x>`).

- Open the Visual Studio command line, and navigate into the `sslutils` directory. Use the following commands to build sslutils:

```
SET
USE_PGXS=1

SET GETTEXTPATH=\
<path_to_gettext>

SET OPENSSLPATH=\
<path_to_openssl>

SET PGPATH=\
<path_to_pg_installation_dir>

SET ARCH=x86

msbuild sslutils.proj
/p:Configuration=Release
```

Where:

- `path_to_gettext` specifies the location of the `GETTEXT` library and header files.
- `path_to_openssl` specifies the location of the openssl library and header files.
- `path_to_pg_installation_dir` specifies the location of the Postgres installation.

For example, the following set of commands builds OpenSSL support into the PostgreSQL 11 server:

```
```.ini
SET USE_PGXS=1

SET OPENSSLPATH=C:\OpenSSL-Win32

SET GETTEXTPATH="C:\Program Files\PostgreSQL\11"

SET PGPATH="C:\Program Files\PostgreSQL\11"

SET ARCH=x86

msbuild sslutils.proj /p:Configuration=Release
```
```

- When the build completes, the `sslutils` directory contains the following files:

```
sslutils--1.3.sql
```

```
sslutils--unpackaged--1.3.sql
```

```
sslutils--pemagent.sql.in
```

```
sslutils.dll
```

- Copy the compiled sslutils files to the appropriate directory for your installation; for example:

```
COPY sslutils*.sql
"%PGPATH%\share\extension\"
```

```
COPY sslutils.dll "%PGPATH%\lib\"

COPY sslutils.control
"%PGPATH%\share\extension\"
```

3. Stop the services of both the old backend database and the new backend database.

On RHEL 8.x, open a command line and assume the identity of a superuser. Enter the command:

```
systemctl <service_name> stop
```

Where `<service_name>` specifies the name of the Postgres service.

On Windows, you can use the Services dialog box to control the service. To stop the service:

1. On the Control Panel select **System and Security > Administrative Tools**.
1. Double-click **Services**.
1. In the Services dialog box, select the service name and select **Stop**.

4. Use the `pg_upgrade` utility to perform an in-place transfer of existing data between the old backend database and the new backend database. If your server is configured to enforce md5 authentication, you might need to add an entry to the `.pgpass` file that specifies the connection properties (and password) for the database superuser. Or you might need to modify the `pg_hba.conf` file to allow trust connections before invoking `pg_upgrade`. For more information about creating an entry in the `.pgpass` file, see the [PostgreSQL core documentation](#).

During the upgrade process, `pg_upgrade` writes a series of log files. The cluster owner must invoke `pg_upgrade` from a directory in which they have write privileges. If the upgrade completes successfully, `pg_upgrade` removes the log files when the upgrade completes. If you don't want `pg_upgrade` to delete the upgrade log files, include the `--retain` keyword when invoking `pg_upgrade`.

To invoke `pg_upgrade`, assume the identity of the cluster owner, navigate to a directory in which the cluster owner has write privileges, and execute the command:

```
<path_to_pg_upgrade> pg_upgrade

-d <old_data_dir_path>

-D <new_data_dir_path>

-b <old_bin_dir_path> -B <new_bin_dir_path>

-p <old_port> -P <new_port>

-u <user_name>
```

Where:

- `path_to_pg_upgrade` specifies the location of the `pg_upgrade` utility. By default, `pg_upgrade` is installed in the `bin` directory under your Postgres directory.
- `old_data_dir_path` specifies the complete path to the data directory of the old backend database.
- `new_data_dir_path` specifies the complete path to the data directory of the new backend database.
- `old_bin_dir_path` specifies the complete path to the bin directory of the old backend database.
- `new_bin_dir_path` specifies the complete path to the bin directory of the old backend database.
- `old_port` specifies the port on which the old server is listening.
- `new_port` specifies the port on which the new server is listening.
- `user_name` specifies the name of the cluster owner.

For example, the following command instructs `pg_upgrade` to migrate the PEM database from PostgreSQL 9.6 to PostgreSQL 11 on a Windows

system (if the backend databases are installed in their default locations):

```
C:\>"C:\Program Files\PostgreSQL\11\bin\pg_upgrade.exe"

-d "C:\Program Files\PostgreSQL\10\data"

-D "C:\Program Files\PostgreSQL\11\data"

-b "C:\Program Files\PostgreSQL\10\bin"

-B "C:\Program Files\PostgreSQL\11\bin"

-p 5432 -P 5433

-U postgres
```

Once invoked, pg\_upgrade performs consistency checks before moving the data to the new backend database. When the upgrade is finished, pg\_upgrade notifies you that the upgrade is complete.

For detailed information about using pg\_upgrade options or troubleshooting the upgrade process, see [pg\\_upgrade](#).

- 5. Copy the following certificate files from the `data` directory of the old backend database to the `data` directory of the new backend database:

- `ca_certificate.crt`
- `ca_key.key`
- `root.crt`
- `root.crl`
- `server.key`
- `server.crt`

Once in place on the target server, make sure the files have these platform-specific permissions:

Permissions and ownership on Linux

| File name          | Owner    | Permissions |
|--------------------|----------|-------------|
| ca_certificate.crt | postgres | -rw-----    |
| ca_key.key         | postgres | -rw-----    |
| root.crt           | postgres | -rw-----    |
| root.crl           | postgres | -rw-----    |
| server.key         | postgres | -rw-----    |
| server.crt         | postgres | -rw-r--r--  |

On Linux, the certificate files must be owned by postgres. You can use the following command at the command line to modify the ownership of the files:

```
chown postgres <file_name>
```

Where `file_name` specifies the name of the certificate file.

Only the owner of the `server.crt` file can modify it, but any user can read it. You can use the following command to set the file permissions for the `server.crt` file:

```
chmod 644 server.crt
```

Only the owner of the other certificate files can modify or read the file. You can use the following command to set the file permissions:

```
chmod 600 <file_name>
```

Where `file_name` specifies the name of the file.

### Permissions and ownership on Windows

On Windows, the service account that performed the PEM server and backend database installation on the target host must own the certificate files moved from the source host. If you invoked the PEM server and Postgres installer using **Run as Administrator** from the context menu of the installer, the owner of the certificate files is Administrators.

To review and modify file permissions on Windows, right-click the file name and select **Properties**.

On the **Security** tab select a group or user name to view the assigned permissions. Select **Edit** or **Advanced** to open dialog boxes that allow you to modify the permissions associated with the selected user.

- The `postgresql.conf` file contains parameter settings that specify server behavior. Modify the `postgresql.conf` file on the new server to match the configuration specified in the `postgresql.conf` file of the old server.

By default, the `postgresql.conf` file is located:

- For Postgres version earlier than 10 on Linux, in `/opt/PostgreSQL/<X>/data`
- For Postgres version 10 or later when installed with graphical installers on Linux, in `/opt/PostgreSQL/<X>/data`
- For Postgres version 10 or later when installed with an RPM on Linux, in `/usr/pgsql/<X>/data`
- For any Postgres version on Windows, in `C:\Program Files\PostgreSQL\<X>\data`

Where `<X>` is the version of Postgres on your system.

Use your choice of editor to update the `postgresql.conf` file of the new server. Modify the following parameters:

- The `port` parameter to listen on the port monitored by your original backend database (typically set to `5432`).
- The `ssl` parameter to be set to `on`

You must also ensure that the following parameters are enabled. If the parameters are commented out, remove the pound sign from in front of each `postgresql.conf` file entry:

- `ssl_cert_file = 'server.crt' # (change requires restart)`
- `ssl_key_file = 'server.key' # (change requires restart)`
- `ssl_ca_file = 'root.crt' # (change requires restart)`
- `ssl_crl_file = 'root.crl'`

Your installation might have other parameter settings that require modification to ensure that the new backend database behaves like the old backend database. Review the `postgresql.conf` files carefully to ensure that the configuration of the new server matches the configuration of the old server.

- The `pg_hba.conf` file contains parameter settings that specify how the server enforces host-based authentication. When you install the PEM server, the installer modifies the `pg_hba.conf` file, adding entries to the top of the file:

```
# Adding entries for PEM Agens and admins to connect to PEM server
```



```
hostssl pem +pem_user 192.168.2.0/24 md5
```

```
hostssl pem +pem_agent 192.168.2.0/24 cert
```

```
# Adding entries (localhost) for PEM Agens and admins to connect to PEM server
```

```
hostssl pem +pem_user 127.0.0.1/32 md5
```

```
hostssl postgres +pem_user 127.0.0.1/32 md5
```

```
hostssl pem +pem_user 127.0.0.1/32 md5
```

```
hostssl pem +pem_agent 127.0.0.1/32 cert
```

By default, the `pg_hba.conf` file is located at the following location:

- o For Postgres version earlier than 10 on Linux, in `/opt/PostgreSQL/<X>/data`
- o For Postgres version 10 or later when installed with graphical installers on Linux, in `/Opt/PostgreSQL/<X>/data`
- o For Postgres version 10 or later when installed with RPMs on Linux, in `/var/lib/pgsql/<X>/data`
- o For Advanced Server version 10 or later when installed with RPMs on Linux, in `/var/lib/edb/as<X>/data`
- o For any Postgres version on Windows, in `C:\Program Files\PostgreSQL\<X>\data`

Where `<X>` is the version of Postgres on your system.

Using your editor of choice, copy the entries from the `pg_hba.conf` file of the old server to the `pg_hba.conf` file for the new server.

8. Restart the service of the new backend database.

On RHEL 8.x, at the command line as superuser enter:

```
systemctl stop <service_name>
```

Where `service_name` is the name of the backend database server.

If you're using Windows, you can use the **Services** dialog box to control the service:

1. In the Control Panel, select **System and Security > Administrative Tools**.
2. Double-click the **Services** icon.
3. In the Services dialog box, select the service name and start the service.

## 9.3 Upgrading SQL Profiler

The process of upgrading SQL Profiler is platform specific. You can update SQL Profiler on a Linux platform by using the native packages and on a Windows platform by using the graphical installer available for Windows.

### Note

- Prior to PEM version 7.8 release, you can install SQL Profiler on Linux either by using the graphical installer or by using the RPMs.
- From PEM version 7.8 and later, SQL Profiler graphical installers for Linux are discontinued. To upgrade a SQL Profiler on a Linux host from any earlier version to PEM 7.9 or later versions, you must use native packages.

Links to SQL Profiler installers and RPMs are available at the [EDB website](#).

### 9.3.1 Upgrading a SQL profiler native package installation on a Linux host

To upgrade a SQL Profiler installation that resides on a Linux host:

1. Delete the existing SQL Profiler query set on each node by invoking the `uninstall-sql-profiler.sql` script. By default, on a Linux host the script resides in the `share/contrib` directory under your EDB Postgres Advanced Server or PostgreSQL installation.

You can use the following server-specific command:

For PostgreSQL:

```
/usr/pgsql-<x>/bin/psql -f /usr/pgsql-<x>/share/contrib/uninstall-sql-profiler.sql -d postgres -U postgres
```

Where `x` is the version of PostgreSQL and `-d` specifies the name of the maintenance database.

For EDB Postgres Advanced Server:

```
/usr/edb/as<x>/bin/psql -f /usr/edb/as<x>/share/contrib/uninstall-sql-profiler.sql -d edb -U enterprisedb
```

Where `x` is the version of EDB Postgres Advanced Server and `-d` specifies the name of the maintenance database.

2. Invoke the new SQL Profiler installer on each node you want to profile.

For PostgreSQL:

```
yum upgrade postgresql<x>-sqlprofiler
```

Where `x` is the version of the PostgreSQL.

For EDB Postgres Advanced Server:

```
yum upgrade edb-as<x>-server-sqlprofiler
```

Where `x` is the version of EDB Postgres Advanced Server.

The installer detects the existing SQL Profiler installation and upgrades with the latest version of SQL Profiler.

See the following example of upgrading SQL Profiler for PostgreSQL:

```
[root@localhost Downloads]# yum install postgresql12-sqlprofiler
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirrors.piconets.webwerks.in
 * epel: hkg.mirror.rackspace.com
 * extras: mirrors.piconets.webwerks.in
 * updates: mirrors.piconets.webwerks.in
Resolving Dependencies
--> Running transaction check
--> Package postgresql12-sqlprofiler.x86_64 0:7.12.0-1.rhel7 will be updated
--> Package postgresql12-sqlprofiler.x86_64 0:7.14.0-1.rhel7 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                                Arch                                Version                                Repository                                Size
=====
Updating:
postgresql12-sqlprofiler                x86_64                                7.14.0-1.rhel7                                edb                                88 k
=====
Transaction Summary
=====
Upgrade 1 Package

Total download size: 88 k
Is this ok [y/d/N]: y
Downloading packages:
No Presto metadata available for edb
postgresql12-sqlprofiler-7.14.0-1.rhel7.x86_64.rpm                                | 88 kB 00:00:04
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Updating : postgresql12-sqlprofiler-7.14.0-1.rhel7.x86_64                                1/2
  Cleanup : postgresql12-sqlprofiler-7.12.0-1.rhel7.x86_64                                2/2
  Verifying : postgresql12-sqlprofiler-7.14.0-1.rhel7.x86_64                                1/2
  Verifying : postgresql12-sqlprofiler-7.12.0-1.rhel7.x86_64                                2/2

Updated:
  postgresql12-sqlprofiler.x86_64 0:7.14.0-1.rhel7

Complete!
```

- Run the `sql-profiler.sql` script file in the maintenance database.

For PostgreSQL:

```
/usr/pgsql-<x>/bin/psql -f /usr/pgsql-<x>/share/contrib/sql-profiler.sql -d postgres -U postgres
```

Where `x` is the version of PostgreSQL and `-d` specifies the name of the maintenance database.

For EDB Postgres Advanced Server:

```
/usr/edb/as<x>/bin/psql -f /usr/edb/as<x>/share/contrib/sql-profiler.sql -d edb -U enterprisedb
```

Where `x` is the version of Advanced Server and `-d` specifies the name of the maintenance database.

- Restart PostgreSQL/EDB Postgres Advanced Server to resume profiling the node from the PEM client.

## 9.3.2 Upgrading a SQL Profiler installation on a Windows host

If you're using SQL Profiler on a Windows host, Windows locks any files that executed or were loaded into memory. To release any locked files, you must stop the Postgres server before performing an upgrade.

On Windows, you can use the Services dialog box to control the service. To stop the service:

- In the Windows Control Panel, select **System and Security > Administrative Tools**.
- Double-click **Services**.
- In the Services dialog box, select the service name, and select **Stop**.

After stopping the Postgres Server:

1. Delete the existing SQL Profiler query set on each node by invoking the `uninstall-sql-profiler.sql` script. By default, on a Windows host the script resides in the `share\contrib` directory under your EDB Postgres Advanced Server or PostgreSQL installation.

You can use the following server-specific commands.

For PostgreSQL:

```
psql -f C:\Program Files\PostgreSQL\<x>\share\contrib\uninstall-sql-profiler.sql -d postgres -U postgres
```

Where `x` is the version of PostgreSQL and `-d` specifies the name of the maintenance database.

For EDB Postgres Advanced Server:

```
psql -f C:\Program Files\edb\as<x>\share\contrib\uninstall-sql-profiler.sql -d edb -U enterprisedb
```

Where `x` is the version of EDB Postgres Advanced Server and `-d` specifies the name of the maintenance database.

2. Invoke the new SQL Profiler installer on each node you want to profile. Run the installer as an Administrator.

For PostgreSQL:

```
sqlprofiler-pg-<x>-<y>-windows-x64.exe
```

Where `x` is the version of the PostgreSQL and `y` is the version of SQL Profiler, for example: `sqlprofiler-pg-12-7.14.0-1-windows-x64.exe`.

For EDB Postgres Advanced Server:

```
sqlprofiler-edb-as<x>-<y>-windows-x64.exe
```

Where `x` is the version of EDB Postgres Advanced Server and `y` is the version of SQL Profiler, for example: `sqlprofiler-edb-as12-7.14.0-1-windows-x64.exe`.

The SQL Profiler installer detects the existing SQL Profiler installation and upgrades it with the latest version of SQL Profiler.

3. Run the `sql-profiler.sql` script file in the maintenance database.

For PostgreSQL:

```
psql -f C:\Program Files\PostgreSQL\<x>\share\contrib\sql-profiler.sql -d postgres -U postgres
```

Where `x` is the version of PostgreSQL and `-d` specifies the name of the maintenance database.

For EDB Postgres Advanced Server:

```
psql -f C:\Program Files\edb\as<x>\share\contrib\sql-profiler.sql -d edb -U enterprisedb
```

Where `x` is the version of EDB Postgres Advanced Server and `-d` specifies the name of the maintenance database.

4. Then, restart the Postgres server to resume profiling the node from a PEM client.

## 9.4 Moving the PEM server

You can move a PEM server from one host machine to a new host machine. The PEM server on the new host (the target) must be installed with the same version of the PEM server installer as the original host (the source). If you don't use the same installer version, you might encounter a schema-mismatch error.

The backend database of the target server (either PostgreSQL or EDB Postgres Advanced Server) can have the same type and version or a different type and version from the backend database of the source PEM server. You can migrate a PEM server that resides on a PostgreSQL host to an EDB Postgres Advanced Server host and vice versa.

Before starting the server migration, make sure that the firewalls between the source host, the target host, and the host of any PEM agent allows connections between the services.

### 1. Prepare the target host.

Invoke the installer for the PEM server on the target host. You must use the same version of the PEM server installer that you used when installing the source PEM server.

The backend database of the target server can have a different version or type from the backend database of the source. If the new PEM server doesn't reside on the same type of backend database as the original server, you must ensure that the same version of the sslutils extension is installed on the new server host. The version of sslutils that's distributed with the PEM installers is freely available for download from the [EDB website](#).

For information about installing the PEM server or the sslutils extension, see the [PEM installation steps](#).

### 2. Drop existing schemas from the new PEM server.

The migration process re-creates the `pem`, `pemdata`, and `pemhistory` schemas from the source PEM server on the target PEM server. To prepare for the move, use the psql client to delete these schemas from the `pem` database on the target host. You can open the psql client at the command line or by selecting **Postgres Enterprise Manager > SQL Shell (psql)**.

When the psql client opens, connect to the `pem` backend database as the database superuser. After connecting to the `pem` database on the target host, drop the schemas:

```
DROP SCHEMA pem
CASCADE;

DROP SCHEMA pemdata CASCADE;

DROP SCHEMA pemhistory CASCADE;
```

When dropping the schemas, you must include the `CASCADE` keyword, instructing the server to delete all dependent objects. When executing the command, the psql client displays a list of the dependent objects. The client confirms each the schema is removed by displaying `DROP SCHEMA`.

### 3. Prepare the PEM agents on the new PEM server.

Before moving the PEM server, you must identify the number of agents that are monitored by the source PEM server and create identities for that number of agents, less one, on the target server. To discover the total number of PEM agents monitored by the PEM server, connect to the `pem` database on the source host with the psql client, and query the `pem.agent` table.

```
SELECT id FROM pem.agent WHERE active =
true;
```

You must manually create the number of agents that reside on the original PEM server, less one. (The PEM server installer creates one agent on the target host.) For example, if the source server contains three agents, you must manually create two more agents. Open a psql session with the `pem` database on the target server, and create the required agents:

```
CREATE USER agent<X>;
```

Where `<X>` specifies an agent number. `agent1` is created on the target host by the PEM server installer.

Then, use the `GRANT` command to assign each agent that resides on the target PEM server `pem_agent` permissions:

```
GRANT pem_agent TO agent<X>;
```

Where `<X>` specifies an agent number.

#### 4. Generate a backup script of the source PEM server.

You can use the `pg_dump` utility to generate a script that contains the commands required to re-create the `pem` database on the target host. By default, `pg_dump` is installed in the `bin` directory under your Postgres installation. To invoke `pg_dump`, in the `bin` directory, enter:

```
pg_dump -U <user_name> <db_name> > <file_name>
```

Where:

- `<user_name>` specifies the name of the database superuser for the PEM backend database.
- `<db_name>` specifies the name of the PEM backend database.
- `<file_name>` specifies the name of the script generated by `pg_dump`.

When prompted, provide the password associated with the user specified.

The command shown instructs `pg_dump` to generate a script that, when executed, re-creates the `pem` database. The script is named `backup.sql` and is created in the `tmp` directory. `pg_dump` is connecting to the server using the credentials of the user `postgres`.

Invoking the `pg_dump` utility doesn't interrupt current database users.

#### Note

If the source PEM server is earlier than the 7.16 version, then you need to replace the following functions before you run `pg_dump` to take backup:

- The `abstime`, `reltime`, and `tinterval` datatypes are deprecated from Postgres version 12 or later. To replace those datatypes with `timestampz` data type, use this command:

```
DO
$$
DECLARE
    rec
record;
    cnt
integer;
BEGIN
    -- Check for the deprecated type in our user info
    probe
    SELECT count(*) INTO
    cnt
    FROM
    pem.probe_column
    WHERE sql_data_type = 'abstime' AND internal_name =
    'valuntil';
    IF cnt = 0
    THEN
        RETURN;
    END IF;
    ALTER TABLE pemdata.user_info
```

```

        ALTER COLUMN valuntil SET DATA TYPE
timestampz;
    ALTER TABLE pemhistory.user_info
        ALTER COLUMN valuntil SET DATA TYPE
timestampz;
    -- Now update the pem.probe_column
    itself
    UPDATE
pem.probe_column
    SET sql_data_type = 'timestampz'
    WHERE sql_data_type = 'abstime' AND internal_name =
'valuntil';
END;
$$ LANGUAGE
'plpgsql';

```

- Replace the this function to avoid any alert errors:

```

CREATE OR REPLACE FUNCTION
pem.check_alert_params_array_size(
template_id pem.alert_template.id%type, params
text[]
)
RETURNS bool AS
$FUNC$
DECLARE
    res bool :=
TRUE;
BEGIN
    /*
    * During restoring the pem database, it does not maintain the order
while
    * inserting data in the table, and uses the sort table based on
the
    *
names.
    * Hence - we need to check the foreign key constraint is present
before
    * validating these
values.
    */
    IF EXISTS(
        SELECT 1 FROM information_schema.table_constraints
        WHERE constraint_name='alert_template_id_fkey' AND
        table_name='alert' AND table_schema='pem'
    )
    THEN
        /*
        * Need to use the IS TRUE construct outside the main query,
because
        * otherwise if there's no template by that ID then the query would
return
        * 0 rows and the result of the function would be undefined and
CHECK
        * constraint would
succeed.
        * Probably this is being over-cautious, because
pem.alert.template_id
        * references pem.alert_template.id. But the SQL standard (probably)
does
        * not define the order in which the CHECK or the FOREIGN KEY
constraints
        * should be validated; in case CHECK is validated first, we want it
to
        *
fail.
        */

```

```
EXECUTE $$SQL$
SELECT
(
SELECT pem.check_array_size_equal(t.param_names,
$2)
FROM pem.alert_template AS
t
WHERE id =
$1
) IS
TRUE
$$SQL$ INTO res USING template_id,
params;
END IF;
RETURN res;
END
$FUNC$ LANGUAGE
'plpgsql';
```

5. Move the backup to the target host.

Move the script generated by the pg\_dump utility to the target host of the PEM server.

6. Restore the backup on the target host.

On the target host, in the `bin` directory under the Postgres backend database installation directory, start psql, executing the script generated by the pg\_dump utility:

```
psql -U <user_name> -d pem -f <file_name>
```

Where:

- `<user_name>` specifies the name of the database superuser. The user specified must have connection privileges for the backend database.
- `<file_name>` specifies the complete path to the backup script generated by pg\_dump.

When prompted, provide the password associated with the database superuser.

The example shown uses the psql client to invoke a script named `backup.sql` to recreate the `pem` database. The script is invoked using the privileges associated with the database superuser postgres.

7. Stop the database server on the target host.

To stop the PEM server on RHEL 8.x, use the command:

```
systemctl stop <service_name>
```

Where `<service_name>` specifies the name of the backend database server. For a PostgreSQL backend database, the service name is `postgresql-<x>`. For an EDB Postgres Advanced Server backend database, the service name is `edb-as-<x>`, where `<x>` specifies the version number.

If you're using Windows, you can use the Services dialog box to control the service. To open the Services dialog box, from the Control Panel, select **System and Security > Administrative Tools**. Double-click the **Services** icon. In the Services dialog box, select the service name in the list, and select **Stop**.

8. Copy the certificate files to the target host.

You must replace the certificate files that are created when the target host is installed with the certificate files of the source host. Copy the following files from the source PEM server to the target PEM server:



- o ca\_certificate.crt
- o ca\_key.key
- o root.crt
- o root.crl
- o server.key
- o server.crt

Copy the files to the `data` directory under the Postgres installation that provides the backend database for the target cluster.

On Linux, the files reside in:

```
/var/lib/pgsql/<X>/data/
```

On Windows, the files reside in:

```
C:\Program Files\PostgreSQL\<X>\data
```

Where:

`<X>` specifies the version of PostgreSQL on your system.

The files already exist on the target cluster. Delete the existing files before performing the copy, or overwrite the existing files with the files from the source server. Once in place on the target server, the files must have the platform-specific permissions shown.

On Linux

| File name          | Owner    | Permissions |
|--------------------|----------|-------------|
| ca_certificate.crt | postgres | -rw-----    |
| ca_key.key         | postgres | -rw-----    |
| root.crt           | postgres | -rw-----    |
| root.crl           | postgres | -rw-----    |
| server.key         | postgres | -rw-----    |
| server.crt         | postgres | -rw-r--r--  |

On Linux, the certificate files must be owned by postgres. Use the following command to modify the ownership of the files:

```
chown postgres <file_name>
```

Where `file_name` specifies the name of the certificate file.

Only the owner of the `server.crt` file can modify the file, but any user can read it. Use the following command to set the file permissions for the `server.crt` file:

```
chmod 644 server.crt
```

Only the owner of the other certificate files can modify or read the files. Use the following command to set the file permissions:

```
chmod 600 <file_name>
```

Where `file_name` specifies the name of the file.

On Windows

On Windows, the service account that performed the PEM server and backend database installation on the target host must own the certificate files moved from the source host. If you invoked the PEM server and Postgres installer using **Run as Administrator** from the installer context menu, the owner of the certificate files is Administrators.

To review and modify file permissions on Windows, right-click the file name and select **Properties**. On the **Security** tab, select a group or user name to view the assigned permissions. Select **Edit** or **Advanced** to open dialog boxes that allow you to modify the permissions associated with the selected user.

- 9. Move the PEM agent certificate files to the PEM server host.

You must move the certificate files used by the PEM agent of the source PEM server to the target host. This step is platform specific.

On Linux

Copy the `agent1.key` and `agent1.crt` files from the source host to the target host. By default, on Linux, the files are installed in `/root/.pem`. Copy the files to the same directory on the target host.

File ownership and permissions of the files must be set to:

| File name  | Owner | Permissions |
|------------|-------|-------------|
| agent1.key | root  | -rw-----    |
| agent1.crt | root  | -rw-r--r--  |

If necessary, navigate to `/root/.pem`, and use the following commands to modify the permissions and ownership of the `agent1.key` file:

```
chmod 600 agent1.key
chown root agent1.key
```

Use the following commands to modify the permissions and ownership of the `agent1.crt` file:

```
chmod 644 agent1.crt
chown root agent1.crt
```

On Windows

Copy the `agent1.key` and `agent1.crt` files from the source host to the target host. On Windows, the files are located in:

```
C:\Users\<user_name>\AppData\Roaming\pem
```

Where `user_name` is the name of the user that invoked the PEM installer.

The ownership and permissions associated with the certificate files on the target machine must match the ownership and permissions of the certificate files on the source machine. If you invoked the PEM server and Postgres installer using **Run as Administrator** on the installer context menu, the owner of the agent certificate files is Administrators.

To review and modify file permissions on Windows, right-click the file name and select **Properties**. On the **Security** tab, select a group or user name to view the assigned permissions. Select **Edit** or **Advanced** to open dialog boxes that allow you to modify the permissions associated with the selected user.

10. Update the `pg_hba.conf` files on the target host.

Modify the `pg_hba.conf` file on the target host to allow connections from each PEM agent. By default, the `pg_hba.conf` file is located in the data directory under your Postgres installation.

11. Start the server on the target host.

After modifying the `pg_hba.conf` file, you must restart the server for the changes to take effect.

To restart the database server on Linux, use the command:

```
/etc/init.d/<service_name> start
```

Where `service_name` is the name of the backend database server.

On Windows, you can use the Services dialog box to control the service. To open the Services dialog box, on the Control Panel, select **System and Security > Administrative Tools**. Double-click the **Services** icon. When the Services dialog box opens, select the service name in the list, and start the service.

12. Connecting monitored agents to the new PEM server host.

To instruct existing PEM agents to connect to the new PEM server host, you must:

- Ensure that the PEM agent host can connect to the new PEM server host.
- Modify the registry on each Windows host with a PEM agent or the agent configuration files on each Linux host with a PEM agent, specifying the IP address and port of the new PEM server.
- Restart the PEM agent's service. These steps are platform specific:
  - [On Linux](#)
  - [On Windows](#)

#### If the PEM agent resides on Linux

Use your choice of editor to modify the `agent.cfg` file, specifying the new IP address and port number of the PEM server in the `pem_host` and `pem_port` parameters.

By default, the `agent.cfg` file is located in:

```
/usr/edb/pem/agent/etc/agent.cfg
```

```

agent.cfg (/opt/PEM/agent/etc) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
agent.cfg
[[PEM/agent]
pem_host=192.168.2.139
pem_port=5432
agent_id=2
agent_ssl_key=/root/.pem/agent2.key
agent_ssl_crt=/root/.pem/agent2.crt
log_level=warning
log_location=/var/log/pem/worker.log
agent_log_location=/var/log/pem/agent.log
long_wait=30
short_wait=10
alert_threads=0
enable_smtp=false
enable_snmp=false
enable_nagios=false
allow_server_restart=true
allow_package_management=false
max_connections=0
connection_lifetime=0
allow_batch_probes=false
batch_script_dir=/tmp
heartbeat_connection=false
allow_streaming_replication=false
ca_file=/opt/PEM/agent/share/certs/ca-bundle.crt
Plain Text Tab Width: 8 Ln 1, Col 1 INS

```

After modifying the `agent.cfg` file, you must restart the PEM agent service. You can use the `pemagent` service script on the Linux command line to restart the service:

```
/etc/init.d/pemagent restart
```

#### If the PEM agent resides on Windows

Before modifying the Windows registry on the monitored node, confirm that the firewall on the host of the PEM agent allows connections to the PEM server. After confirming that the PEM agent host can connect to the PEM server host, you can use the Windows Registry Editor to review and edit the `PEM_HOST` and `PEM_PORT` entries to ensure that they correctly identify the host and port used by the PEM server. To open the Registry Editor, enter `regedit` in the Windows Run dialog box or in the Windows start menu search box. Navigate through the registry tree control to view or modify registry entries.

The PEM agent registry entries are located at: `HKEY_LOCAL_MACHINE\Software\EnterpriseDB\PEM\agent`

The `PEM_HOST` and `PEM_PORT` entries must specify the address and port number of the new PEM server on the target host. To modify a registry entry, right-click the entry name and select **Modify** from the context menu. Then use the Edit String dialog box to make any changes to the value of the entry. After you finish, select **OK**.

After modifying the registry, you must restart the PEM agent's service. You can use the Services dialog box, accessed through the Windows Control Panel, to restart the `Postgres Enterprise Manager - pemagent` service.

After moving the server, change the connection properties in any installed PEM clients to connect to the new host of the PEM server, agents, and monitored servers.

#### Note

After moving the server, if you encounter this error while creating the server in the PEM web interface:

Error - User does not have enough permission to add new server.  
Please contact the administrator to grant 'pem\_database\_server\_registration' role to the 'enterprisedb' user.

Resolve the error by updating the roles and granting appropriate permissions:

```
UPDATE pem.roles SET rolid = pr.oid FROM pg_roles pr WHERE pr.rolname = 'pem_' || component;
```

## 10 Uninstalling

To uninstall the PEM components see,

- [Uninstalling on Linux](#)
- [Uninstalling on Windows](#)

### 10.1 Uninstalling Postgres Enterprise Manager components on Linux

The process of uninstalling the PEM server or Agent is platform-specific. The name of the package for PEM server is `edb-pem-server` and for PEM Agent is `edb-pem-agent`.

If you uninstall the PEM server package from a host, the PEM Agent package installed on the same host doesn't get uninstalled. But if you uninstall the PEM Agent package, then the PEM server package installed on the same host also gets uninstalled.

#### Note

Before uninstalling the PEM Agent, you need to de-register the agent first. You can de-register the agent using the `pemworker` command-line utility. After that, you can proceed with the uninstallation steps.

#### Uninstalling PEM components from RHEL or Rocky Linux or AlmaLinux hosts

You can use variations of the `rpm` command to remove the installed packages from Rocky Linux/AlmaLinux/RHEL 8.x hosts. Also, you can use the `dnf remove` command to remove the installed package from Rocky Linux/AlmaLinux/RHEL 8.x hosts. Note that removing a package does not damage the PEM data directory.

- Include the `-e` option when invoking the `rpm` command to remove an installed package; the command syntax is:

```
rpm -e package_name
```

Where `package_name` is the name of the package that you would like to remove.

- You can use `dnf remove` command to remove the pem server or Agent along with the `edb-pem` and `edb-pem-docs` dependencies on Rocky Linux or AlmaLinux or RHEL 8.x hosts. To remove a package, open a terminal window, assume superuser privileges, and enter the command:

```
dnf remove package_name
```

## Uninstalling PEM components from Debian or Ubuntu hosts

You can use `apt-get remove` or `apt-get purge` command to uninstall the PEM server or Agent package from a Debian or Ubuntu host:

- To uninstall PEM server or Agent from a Debian or Ubuntu host without impacting the configuration files and data directories, invoke the following command:

```
apt-get remove package_name
```

- To uninstall PEM server or Agent along with the configuration files and data directory, invoke the following command:

```
apt-get purge package_name
```

Where `package_name` is the name of the package that you would like to remove.

## Uninstalling PEM components from SLES hosts

To uninstall PEM server or Agent from a SLES host, invoke the following command:

```
zypper remove package_name
```

Where `package_name` is the name of the package that you would like to remove.

## 10.2 Uninstalling Postgres Enterprise Manager components on Windows

If you uninstall the PEM server from a host, the PEM agent installed on the same host is uninstalled. But if you uninstall the PEM agent, then the PEM server installed on the same host will not be uninstalled.

You can use the Windows `Add/Remove Programs` application to remove PEM components from a Windows host. Select the `Add/Remove Programs` option from the Windows `Control Panel`. When the `control panel` opens, locate the name of the PEM component in the program list. Click the `Remove` button to remove the component.

You can also invoke the uninstaller that resides at the following location:

For the PEM Server, `C:\Program Files\edb\pem\server\uninstall-pemserver`

For the PEM Agent, `C:\Program Files\edb\pem\agent\uninstall-pemagent`

## 11 Troubleshooting server installation

### Server installation errors

While installing the PEM server on RHEL 8, you might see this error:

```
[root@etpgxlt firstuser]# dnf install edb-pem
Updating Subscription Management repositories.
Last metadata expiration check: 0:01:33 ago on Wed 30 Mar 2022 01:28:16 AM EDT.
Error:
  Problem: problem with installed package python3-mod_wsgi-4.6.4-4.el8.s390x
    - package python3-mod_wsgi-4.7.1-4.module+el8.4.0+9822+20bf1249.s390x conflicts with python3-mod_wsgi
      provided by python3-mod_wsgi-4.6.4-4.el8.s390x
    - package python3-mod_wsgi-4.7.1-4.module+el8.4.0+9822+20bf1249.s390x conflicts with python3-mod_wsgi
      provided by python3-mod_wsgi-4.6.4-3.el8.s390x
    - package edb-pem-server-8.4.0-7.rhel8.s390x requires python3-mod_wsgi >= 4.7, but none of the providers
      can be installed
    - package edb-pem-8.4.0-7.rhel8.s390x requires edb-pem-server = 8.4.0-7.rhel8, but none of the providers
      can be installed
    - cannot install the best candidate for the job
  (try to add '--allowerase' to command line to replace conflicting packages or '--skip-broken' to skip
  uninstalleable packages or '--nobest' to use not only best candidate packages)
[root@etpgxlt firstuser]#
```

Remove the `python3-mod_wsgi` package first:

```
dnf remove python3-mod_wsgi
```

Then try installing the PEM server again.

## Reconfiguring the PEM server

In some situations, you might need to uninstall the PEM server, reinstall it, and then configure the server again. To do so:

1. Remove the PEM server configuration and uninstall:

```
/usr/edb/pem/bin/configure-pem-server.sh -un
```

2. Remove the PEM packages:

```
yum erase edb-pem-server
```

3. Drop the `pem` database:

```
DROP DATABASE
pem
```

4. Move the certificates from `/root/.pem/` to another location:

```
mv /root/.pem/* <new_location>
```

5. Move the `agent.cfg` file from `/usr/edb/pem/agent/etc/agent.cfg` to another location:

```
mv /usr/edb/pem/agent/etc/agent.cfg <new_location>
```

6. Then, configure the PEM server again:

```
/usr/edb/pem/bin/configure-pem-server.sh
```

## 12 Changing the default port

By default, the 8443 port is assigned for the web services at the time of configuration of the PEM server. You can change the port after configuration by changing a few parameters in the Apache configuration files. The Apache configuration files names and locations are platform specific.

### For RHEL

1. Edit the file `/etc/httpd/conf.d/edb-ssl-pem.conf`, replacing 8443 with your port number in the following parameters:

```
Listen 8443
<VirtualHost _default_:8443>
ServerName localhost:8443
RewriteRule ^(.*)$ https://%{HTTP_HOST}:8443%{REQUEST_URI} [L,R=301]
```

2. If `semanage` isn't installed, install it:

```
sudo yum -y install policycoreutils-python
```

3. Check whether your port is listed:

```
semanage port -l | grep http
```

4. If your port doesn't appear in the list, run this command:

```
sudo semanage port -a -t http_port_t -p tcp <your_port_number>
```

5. Restart the httpd service:

```
sudo systemctl restart httpd
```

You can now access the PEM web interface using your port. For more details, see [Accessing the web interface](#).

### For Debian and Ubuntu

1. Edit the file `/etc/apache2/sites-available/edb-ssl-pem.conf`, replacing 8443 with your port number in the following parameters:

```
Listen 8443
<VirtualHost _default_:8443>
ServerName HTTPD-EDBPEM-SERVER-
v8:8443
```

2. Restart the httpd service:

```
sudo systemctl restart apache2
```

You can now access the PEM web interface using your port. For more details, see [Accessing the web interface](#).



## 13 Registering a Postgres server

Before you can manage or monitor a database server with PEM, you must register the database server with PEM and bind the database server to a previously registered PEM agent. You can bind a database server to a remote agent (an agent that resides on a different host). However, if the agent resides on a different host, it doesn't have access to all of the statistical information about the instance.

### How to register a Postgres server

There are three methods to register a server:

- [Use the pemworker utility to register a database server](#) from the command line on the host where the agent you want to bind is located.
- [Use automatic server discovery](#) in the PEM web application to automatically register Postgres servers located on the same host as an agent.
- [Manually register a Postgres server](#) by adding all the details in the PEM web application.

### What does registering a Postgres server do?

Registering a server does two things:

- It adds the server to the server tree in the PEM web application, allowing users to open connections from the web application to the server.
- It binds the server to a particular agent, meaning that agent will open connections to the server to execute probes and scheduled jobs.

#### Connections from the agent to the monitored server

During agent registration, you're required to provide connection details for the agent to connect to the monitored server. As you normally do for Postgres, you must specify the database to connect to. However, the PEM agent will monitor **every database accessible to the specified user**, not just the database specified.

After registration, the agent opens a connection to the specified database. It uses this connection to determine the other databases it can connect to with the same user and then spawns one connection to each.

### Using the pemworker utility to register a server

You can use the pemworker utility to register a Postgres server. During registration, the pemworker utility binds the new server to the agent that resides on the system from which you invoked the registration command.

To register a server on a Linux host, use the command:

```
pemworker --register-server
```

To register a server on a Windows host, use the command:

```
pemworker.exe REGISTER-SERVER
```

Append command line options to the command string when invoking the pemworker utility. Follow each option with a corresponding value.

| Option | Description |
|--------|-------------|
|--------|-------------|

| Option                                   | Description   |
|--|---|
| <code>--pem-user</code>                  | Specifies the name of the PEM administrative user (must have the <code>pem_admin</code> role) on the PEM server to use to write the server details to the PEM database. Required.   |
| <b>Server parameters</b>                 | These parameters are used to populate the connection properties in the PEM web application. They're used when a user connects from the PEM web application to the monitored server. They are also used for connections from the agent to the monitored server unless overridden by Agent Server Binding parameters (see <b>Agent-server binding parameters</b> in this table).  |
| <code>--server-addr</code>               | Specifies the IP address or fully qualified domain name of the monitored server. On Linux systems, you can leave the address field blank to use the default PostgreSQL Unix Domain Socket on the local machine. Or you can set it to an alternative path containing a PostgreSQL socket. If you enter a path, the path must begin with a forward slash (/). Required.   |
| <code>--server-port</code>               | Specifies the port number of the monitored server. Required.  |
| <code>--server-database</code>           | Specifies the name of the database on the monitored server to which to connect. Required.   |
| <code>--server-user</code>               | Specifies the name of the user used to connect to the monitored server. Required.   |
| <code>--server-service-name</code>       | Specifies the name of the operating system service that manages the monitored Postgres server, for example, a <code>systemd</code> service unit on Linux. Optional.   |
| <b>Agent-server binding parameters</b>   | These parameters are used to override the server parameters. Use these if you want the agent to connect to the monitored server using different credentials from those used by the PEM web application.   |
| <code>--asb-host-name</code>             | Specifies the IP address or fully qualified domain name of the monitored server. Optional, defaults to <code>--server-addr</code> if not supplied.  |
| <code>--asb-host-port</code>             | Specifies the port number of the monitored server. Optional, defaults to <code>--server-port</code> if not supplied.  |
| <code>--asb-host-db</code>               | Specifies the name of the database on the monitored server to which the agent connects. Optional, defaults to <code>--server-database</code> if not supplied.   |
| <code>--asb-host-user</code>             | Specifies the name of the user used by the agent to connect to the monitored server. Optional, defaults to <code>--server-user</code> if not supplied.  |
| <code>--asb-ssl-mode</code>              | Specifies the type of SSL authentication to use for connections. Supported values include: <code>prefer</code> , <code>require</code> , <code>disable</code> , <code>verify-CA</code> , and <code>verify-full</code> . Optional, defaults to <code>prefer</code> .  |
| <b>Server metadata</b>                   | These parameters determine how the server is shown in the PEM web application.  |
| <code>--group</code>                     | Specifies the name of the group in which the server is displayed. Optional.   |
| <code>--team</code>                      | Specifies a Postgres role on the PEM server to be assigned as the team to which the monitored server belongs. Only users with this role can access the server. Optional, defaults to <code>none</code> , meaning all users can access.  |
| <code>--owner</code>                     | Specifies a Postgres user on the PEM server to be assigned as the owner of the monitored server. Optional, defaults to <code>--pem-user</code> .  |
| <code>--display-name &lt;name&gt;</code> | Specifies the display name of the monitored database server. Optional, defaults to the system hostname.   |
| <b>Other parameters</b>                  |   |
| <code>--remote-monitoring</code>         | Set to <code>yes</code> if the server isn't located on the same host as the agent. When remote monitoring is enabled ( <code>yes</code> ), agent level statistics for the monitored server aren't available for custom charts and dashboards, and the remote server isn't accessible by some PEM utilities, such as Audit Manager, Capacity Manager, Log Manager, Postgres Expert, and Tuning Wizard. Optional, defaults to <code>no</code> . |
| <code>--efm-cluster-name</code>          | Specifies the name of the EDB Failover Manager cluster that monitors the server (if applicable). Optional.  |
| <code>--efm-install-path</code>          | Specifies the complete path to the installation directory of EDB Failover Manager (if applicable). Optional.  |
| <code>--config-dir</code>                | Specifies the directory path of the agent configuration file. Optional, defaults to <code>&lt;pemworker path&gt;/../etc</code> .  |

Set the environment variable `PEM_SERVER_PASSWORD` to provide the password for the PEM server to allow the `pemworker` to connect as a PEM admin user.

Set the environment variable `PEM_MONITORED_SERVER_PASSWORD` to provide the password of the database server being registered and monitored by the PEM agent.

If you don't provide the password, a password authentication error occurs. The PEM server acknowledges that the server was registered properly.

### Examples

This example registers a server using only the required parameters. The user `admin01` will be used to connect to the PEM server, and the credentials supplied for the `--server-*` parameters will be used to create a connection from the agent to the monitored server. The same details will also be used to populate the server connection details in the PEM web application.

```
pemworker --register-server \
  --pem-user admin01 \
  --server-addr pg123.prod.infra.business \
  --server-port 5432 \
  --server-database postgres \
  --server-user prod_admin
```

This example specifies different parameters to use for the connection from the agent to the monitored server. The example overrides the fully qualified domain name provided in `--server-addr` with `localhost` instead and overrides the user to use a `local_monitor` user. This might be a user that can connect only from the same host, for example.

Also specified is a service name, `postgresql`, which means the PEM agent can restart this server to apply configuration changes.

```
pemworker --register-server \
  --pem-user admin01 \
  --server-addr pg123.prod.infra.business \
  --server-port 5432 \
  --server-database postgres \
  --server-user prod_admin \
  --server-service-name postgresql \
  --asb-host-name localhost \
  --asb-host-user local_monitor
```

### Using the pemworker utility to unregister a server

You can use the pemworker utility to unregister a database server. To unregister a server, invoke the pemworker utility.

On a Linux host, use the command:

```
pemworker --unregister-server
```

On a Windows host, use the command:

```
pemworker.exe UNREGISTER-SERVER
```

Append command line options to the command string when invoking the pemworker utility. Follow each option with by a corresponding value:

| Option                  | Description  |
|-------------------------|--|
| <code>--pem-user</code> | Specifies the name of the PEM administrative user. Required. |

| Option                                    | Description  |
|---|--|
| <code>--<br/>serv<br/>er-<br/>addr</code> | Specifies the IP address of the server host or the fully qualified domain name. On Unix-based systems, you can leave the address field blank to use the default PostgreSQL Unix Domain Socket on the local machine. Or, you can set it to an alternative path containing a PostgreSQL socket. If you enter a path, the path must begin with a forward slash (/). Required. |
| <code>--<br/>serv<br/>er-<br/>port</code> | Specifies the port number of the host. Required.   |
| <code>--<br/>conf<br/>ig-<br/>dir</code>  | Specifies the directory path of the agent configuration file. The default is <code>"&lt;pemworker path&gt;/../etc"</code> .  |

Use the `PEM_SERVER_PASSWORD` environment variable to provide the password for the PEM server to allow the pemworker to connect as a PEM admin user.

If you don't provide the password, a password authentication error occurs. The PEM server acknowledges that the server is unregistered.

## Registering a server with automatic server discovery

If the server you want to monitor resides on the same host as the monitoring agent, you can use the Auto Discovery dialog box to simplify the registration and binding process.

### Limitations of automatic discovery

Automatic server discovery doesn't perform an exhaustive search for Postgres binaries or services. It detects only Postgres instances created directly by the package installer.

To enable auto discovery for a specific agent, you must enable the Server Auto Discovery probe. To do so, select the PEM agent in the PEM client tree, and select **Management > Manage Probes**. When the **Manage Probes** tab opens, confirm that the slider in the **Enabled?** column is set to **Yes**.

To open the Auto Discovery dialog box, select a PEM agent in the PEM client tree and select **Management > Auto Discovery**.

When the Auto Discovery dialog box opens, the **Discovered Database Servers** box displays a list of servers that currently aren't being monitored by a PEM agent. Select the box next to a server name to display information about the server in the **Server Connection Details** box and connection properties for the agent in the **Agent Connection Details** box.

Use **Check All** to select the box next to all of the displayed servers or **Uncheck All** to clear all of the boxes to the left of the server names.

The fields in the **Server Connection Details** box provide information about the server that PEM monitors:

- Accept or modify the name of the monitored server in the **Name** field. The specified name is displayed in the tree of the PEM client.
- Use the **Server group** list to select the server group under which the server is displayed in the PEM client tree.
- Use the **Host name/address** field to specify the IP address of the monitored server.
- The **Port** field displays the port that's monitored by the server. You can't modify this field.
- Provide the name of the service in the **Service ID** field. You must provide the service name to enable some PEM functionality.
- The **Maintenance database** field specifies which database to use for the initial connection and any global maintenance operations. Customize the content of the **Maintenance database** field for your installation.

The fields in the **Agent Connection Details** box specify the properties for the PEM agent to use when connecting to the server:

- The **Host** field displays the IP address used for the PEM agent binding.

- The **User name** field displays the name used by the PEM agent when connecting to the selected server.
- The **Password** field displays the password associated with the specified user name.
- Use the **SSL mode** field to specify your SSL connection preferences.

After you finish specifying the connection properties for the servers that you're binding for monitoring, select **OK** to register the servers.

After selecting **OK**, the newly registered server is displayed in the PEM tree and is monitored by the PEM server.

## Manually registering a database server

To manage or monitor a database server with PEM, you must:

- Register your EDB Postgres Advanced Server or PostgreSQL server with the PEM server.
- Bind the database server to a PEM agent.

You can use the Create Server dialog box to provide registration information for a server, bind a PEM agent, and display the server in the PEM client tree. To open the Create Server dialog box, select **Object > Create > Server**.

### Note

You must ensure the `pg_hba.conf` file of the Postgres server that you're registering allows connections from the host of the PEM client before attempting to connect.

Use the **General** tab to describe the general properties of the server:

- Use the **Name** field to specify a name for the server. The name identifies the server in the PEM browser tree.
- You can use groups to organize your servers and agents in the tree. Using groups can help you manage large numbers of servers more easily. For example, you can have a production group, a test group, or LAN-specific groups. Use the **Group** list to select the server group in which to display the new server.
- Use the **Team** field to specify a PostgreSQL role name. Only PEM users who are members of this role, who created the server initially, or have superuser privileges on the PEM server see this server when they log in to PEM. If this field is left blank, by default all PEM users see the server. You can use the `show_objects_with_no_team` parameter in the Server Configuration dialog box to change the behavior. If `show_objects_with_no_team` is set to `false`, the server with no team isn't visible to all other users.
- Use the **Background** color selector to select the color to display in the PEM tree behind database objects that are stored on the server.
- Use the **Foreground** color selector to select the font color of labels in the PEM tree for objects stored on the server.
- Select **Connect now?** to attempt a server connection when you select **Save**. Clear **Connect now?** if you don't want the PEM client to validate the specified connection parameters until a later connection attempt.
- Provide notes about the server in the **Comments** field.

Use the **Connection** tab to specify connection details for the server:

- Specify the IP address of the server host or the fully qualified domain name in the **Host name/address** field. On Unix-based systems, leave the address field blank to use the default PostgreSQL Unix Domain Socket on the local machine. Or you can set an alternative path containing a PostgreSQL socket. If you enter a path, the path must begin with a forward slash (/).
- Specify the port number of the host in the **Port** field.
- Use the **Maintenance database** field to specify the name of the initial database for PEM to connect to that's expected to contain `pqAgent` schema and `adminpack` objects installed (both optional). On PostgreSQL version 8.1 and later, the maintenance DB is normally called `postgres`. On earlier versions `template1` is often used, although it's better to create a `postgres` database to avoid cluttering the template database.
- Set **Kerberos Authentication** to **Yes** to use the Kerberos authentication for a monitored server. By default, the monitored server uses the same authentication method as the PEM server. If the monitored server doesn't want to use Kerberos authentication, then set `ALLOW_DATABASE_CONNECTION_WITHOUT_KERBEROS` to `TRUE` in the `config_local.py` file.
- Specify the name to use when authenticating with the server in the **Username** field.
- Provide the password associated with the specified user in the **Password** field.
- Select **Save password?** to store passwords in encrypted format in a PEM backend database for later reuse. Each password is stored on a per-user, per-server basis and isn't shared with other team members. PEM uses the saved password to connect the database server next time. To remove a saved password, disconnect the database server first, and then select **Object > Clear Saved Password**.
- Use the **Role** field to specify the name of the role that's assigned the privileges for the client to use after connecting to the server. This value allows you to connect as one role and then assume the permissions of another role (the one you specified in this field) when the connection is established. The connecting role must be a member of the role specified.

Use the **SSL** tab to configure SSL:

- In the **SSL mode** field, select the type of SSL connection for the server to use. For more information about using SSL encryption, see [the PostgreSQL documentation](#).

You can use the platform-specific file manager dialog box to upload files that support SSL encryption to the server. To open the file manager, select the icon located to the right of each of the following fields:

- Use the **Client certificate** field to specify the file containing the client SSL certificate. This file replaces the default `~/.postgresql/postgresql.crt` file if PEM is installed in Desktop mode and `<STORAGE_DIR>/<USERNAME>/.postgresql/postgresql.crt` if PEM is installed in Web mode. This parameter is ignored if an SSL connection isn't made.
- Use the **Client certificate key** field to specify the file containing the secret key used for the client certificate. This file replaces the default `~/.postgresql/postgresql.key` if PEM is installed in Desktop mode and `<STORAGE_DIR>/<USERNAME>/.postgresql/postgresql.key` if PEM is installed in Web mode. This parameter is ignored if an SSL connection isn't made.
- Use the **Root certificate** field to specify the file containing the SSL certificate authority. This file replaces the default `~/.postgresql/root.crt` file. This parameter is ignored if an SSL connection isn't made.
- Use the **Certificate revocation list** field to specify the file containing the SSL certificate revocation list. This list replaces the default list, found in `~/.postgresql/root.crl`. This parameter is ignored if an SSL connection isn't made.
- When **SSL compression?** is set to **True**, data sent over SSL connections is compressed. The default value is **False** (compression is disabled). This parameter is ignored if an SSL connection isn't made.

### Warning

Certificates, private keys, and the revocation list are stored in the per-user file storage area on the server, which is owned by the user account under which the PEM server process is run. This means that administrators of the server might be able to access those files. Use caution before using this feature.

Use the **SSH Tunnel** tab to configure SSH tunneling. You can use a tunnel to connect a database server through an intermediary proxy host to a server that resides on a network to which the client might not be able to connect directly.

- Set **Use SSH tunneling** to **Yes** to use an SSH tunnel when connecting to the specified server.
- Specify the name or IP address of the SSH host (through which client connections are forwarded) in the **Tunnel host** field.
- Specify the port of the SSH host through which client connections are forwarded in the **Tunnel port** field.
- Specify the name of a user with login privileges for the SSH host in the **Username** field.
- Specify the type of authentication to use when connecting to the SSH host in the **Authentication** field.
- Select **Password** to use a password for authentication to the SSH host. This is the default.
- Select **Identity file** to use a private key file when connecting.
- If the SSH host is expecting a private key file for authentication, use the **Identity file** field to specify the location of the key file.
- If the SSH host is expecting a password, use the **Password** field to specify the password. If an identity file is being used, specify the passphrase.

Use the **Advanced** tab to specify details that are used to manage the server:

- Specify the IP address of the server host in the **Host Address** field.
- Use the **DB restriction** field to specify a SQL restriction to use against the `pg_database` table to limit the databases displayed in the tree. For example, you might enter: `'live_db', 'test_db'` to display only the `live_db` and `test_db` databases. You can also limit the schemas shown in the database from the database Properties dialog box by entering a restriction against `pg_namespace`.
- Use the **Password file** field to specify the location of a password file ( `.pgpass` ). The `.pgpass` file allows a user to log in without providing a password when they connect. The file must be present on the PEM server. For more information, see the [Postgres documentation](#).
- Use the **Service ID** field to specify parameters to the database service process. For servers that are stored in the Enterprise Manager directory, enter the service ID. On Windows machines, this is the identifier for the Windows service. On Linux machines, the name of the init script used to start the server is `/etc/init.d` and the name of the systemd script to start the server is `systemctl`. For example, the name of the EDB Postgres Advanced Server 11 service is `edb-as-11`. For local servers, the setting is operating system dependent:
  - If the PEM client is running on a Windows machine, it can control the postmaster service if you have enough access rights. Enter the name of the service. In case of a remote server, prepend it with the machine name (such as `PSE1\pgsql-8.0`). PEM automatically discovers services running on your local machine.

- If the PEM client is running on a Linux machine, it can control processes running on the local machine if you have enough access rights. Provide a full path and needed options to access the `pg_ctl` program. When executing service control functions, PEM appends status/start/stop keywords to this. For example:

```
sudo /usr/pgsql-x/bin/pg_ctl -D /var/lib/pgsql/x/data
```

Where `x` is the version of the PostgreSQL database server.

- If the server is a member of a Failover Manager cluster, you can use PEM to monitor the health of the cluster and to replace the primary node if necessary. To enable PEM to monitor Failover Manager, use the **EFM cluster name** field to specify the cluster name. The cluster name is the prefix of the name of the Failover Manager cluster properties file. For example, if the cluster properties file is named `efm.properties`, the cluster name is `efm`.
- If you're using PEM to monitor the status of a Failover Manager cluster, use the **EFM installation path** field to specify the location of the Failover Manager binary file. By default, the Failover Manager binary file is installed in `/usr/edb/efm-x.x/bin`, where `x.x` specifies the Failover Manager version.

Use the **PEM Agent** tab to specify connection details for the PEM agent.

On the **Connection Parameters** tab:

- Set **Remote monitoring?** to **Yes** to indicate that the PEM agent doesn't reside on the same host as the monitored server. When remote monitoring is enabled, agent level statistics for the monitored server aren't available for custom charts and dashboards, and the remote server can't be accessed by some PEM utilities (such as Audit Manager, Capacity Manager, Log Manager, Postgres Expert, and Tuning Wizard).
- Use the **Bound agent** list to select a PEM agent. One agent can monitor multiple Postgres servers.
- In the **Host** field, enter the IP address or socket path for the agent to use when connecting to the database server. By default, the agent uses the host address shown on the **General** tab. On a Unix server, you might want to specify a socket path, such as `/tmp`.
- Enter the port number for the agent to use when connecting to the server. By default, the agent uses the port defined on the **Properties** tab.
- Use the **SSL** field to specify an SSL operational mode. Specify **require**, **prefer**, **allow**, **disable**, **verify-ca**, or **verify-full**.

| Mode        | Description  |
|-------------|--|
| require     | Require SSL encryption for transactions between the server and the agent.  |
| prefer      | Use SSL encryption between the server and the agent if SSL encryption is available.  |
| allow       | Allow the connection to use SSL if required by the server.   |
| disable     | Disable SSL encryption between the agent and the server.   |
| verify-ca   | Require SSL encryption and require the server to authenticate using a certificate registered by a certificate authority.         |
| verify-full | Require SSL encryption and require the server to authenticate using a certificate registered by a trusted certificate authority. |

For more information about using SSL encryption, see the [PostgreSQL documentation](#).

- Use the **Database** field to specify the name of the Postgres database to which the agent initially connects. If you're registering a EDB Postgres Distributed database node then specify the EDB Postgres Distributed-enabled database name in this field.
- In the **User name** field, specify the name of the role for the agent to use when connecting to the server. The specified role must be a database superuser for all of the features to work as expected. For the list of features that don't work if the specified role isn't a database superuser, see [Agent privileges](#). If you're using the Kerberos authentication method, then specify the user having the `pgd_monitor` or `pgd_superuser` role in this field.

If you're using Postgres version 10 or later, you can use the `pg_monitor` role to grant the required privileges to a non-superuser. For information about the `pg_monitor` role, see [Default Roles](#).

- In the **User name** field, specify the name of the user for the agent to use when connecting to the server. The specified role must be a database superuser for all of the features to work as expected. If you're using Postgres version 10 or later, you can use the `pg_monitor` role to grant the required privileges to a nonsuperuser. For information about the `pg_monitor` role, see [Default Roles](#).
- Specify the password for the agent to use when connecting to the server in the **Password** field and **Confirm password** fields. If you don't specify a password, you must configure the authentication for the agent manually. For example, you can use a `.pgpass` file, which must be present and accessible on the system where agent is installed.

On the **Advanced** tab:

- Set **Allow takeover?** to **Yes** to specify that another agent can take over the server. This feature allows an agent to take responsibility for the monitoring of the database server if, for example, the server moved to another host as part of a high-availability failover process.
- Use the plus sign (+) to add the database you want to exclude from the PEM monitoring. You can't exclude the database mentioned on the **Connection Parameters** tab of the **PEM Agent** tab.

#### Note

The database-level probes don't execute for excluded databases, but the server-level probes can collect the database statistics.

Use the **General** tab under the **BART** tab to describe the general properties of the BART server that map to the PEM server:

- Use the **BART server** field to select the BART server name. All the BART servers configured in the PEM console are listed.
- Use the **Server name** field to specify a name for the database server that you want to back up using the BART server. This name gets stored in the BART configuration file.
- Use the **Backup name** field to specify a template for user-defined names to assign to the backups of the database server. If you don't specify a backup name template, then you can reference the backup in BART sub-commands only by using the BART-assigned integer backup identifier.
- Use the **Host address** field to specify the IP address of the database server that you want to configure for backup.
- Use the **Port** field to specify the port to use for the database that you want to back up.
- Use the **User** field to specify the user of the database that you want to back up using BART through the PEM console. If you want to enable incremental backups for this database server, then the user must be a superuser.
- Use the **Password** field to specify the password for the user of the database that you want to back up.
- Use the **Cluster Owner** field to specify the Linux operating system user account that owns the database cluster. This is typically `enterprisedb` for EDB Postgres Advanced Server database clusters installed in the Oracle databases compatible mode. It is typically `postgres` for PostgreSQL database clusters and for EDB Postgres Advanced Server database clusters installed in the PostgreSQL databases compatible mode.
- Use the **Archive command** field to specify the desired format of the archive command string to use in the `bart.cfg` file. Inputs provided for the Archive command overwrite the database server's `Postgresql.conf` file. After the server gets added, the database server restarts or database configurations are reloaded.
- Use **Allow incremental backup?** to specify whether to enable incremental backup for this database server.
- Use **Setup passwordless SSH?** to specify if you want to create SSH certificates to allow passwordless logins between the database server and the BART server. Be sure to bind a PEM agent before setting up the passwordless SSH authentication. Passwordless SSH doesn't work for a database server being remotely monitored by a PEM agent.

Use the **Misc** tab under the **BART** tab to describe the miscellaneous properties of the BART Server:

- Use **Override default configuration?** to override the BART server configurations with the specific database server configurations.
- Use the **Xlog** method to specify how to collect the transaction log while `pg_basebackup` is executing.
- Use the **Retention policy** field to specify the retention policy for the backup. This determines when to mark an active backup as obsolete and hence a candidate for deletion. You can specify the retention policy in terms of number of backups or duration (days, weeks, or months).
- Use **WAL compression** to specify if you want to compress the archived Xlog/WAL files in gzip format. To enable WAL compression, the gzip compression program must be present in the BART user account's `PATH`. Don't enable the **WAL compression** setting for database servers for which you need to take incremental backups.
- Use the **Copy WALs during restore** field to specify how the archived WAL files are collected when invoking the RESTORE operation. Set to **Enabled** to copy the archived WAL files from the BART backup catalog to the `<restore_path>/archived_wals` directory prior to the database server archive recovery. Set to **Disabled** to retrieve the archived WAL files directly from the BART backup catalog during the database server archive recovery.
- Use the **Thread count** field to specify the number of threads to copy the blocks. You must set **Thread count** to **1** if you want to take a backup using the `pg_basebackup` utility.
- Use the **Batch size** field to specify the number of blocks of memory used for copying modified blocks. This option applies only to incremental backups.
- Use the **Scan interval** field to specify the number of seconds for the WAL scanner to wait before scanning the new WAL files.
- Use the **MBM scan timeout** field to specify the number of seconds to wait for MBM files before timing out. This option applies only to incremental backups.



To view the properties of a server, right-click the server name in the PEM client tree, and select **Properties** from the context menu. To modify a server's properties, disconnect from the server before opening the Properties dialog box.

## Verifying the connection and binding

Once registered, the new server is added to the PEM browser tree and is displayed on the Global Overview dashboard.

When first connecting to a newly bound server, the Global Overview dashboard might display the new server with a status of “unknown” in the server list. Before recognizing the server, the bound agent must execute a number of probes to examine the server. These probes might take a few minutes to complete, depending on network availability.

In a few minutes, bar graphs on the Global Overview dashboard show that the agent connected successfully. The new server is included in the **Postgres Server Status** list.

If after five minutes the Global Overview dashboard still doesn't list the new server, check the logfiles for the monitoring agent for errors. Right-click the agent's name in the tree, and select the **Dashboards > Probe Log Analysis** from the context menu.

## 14 Registering a PEM agent

Before a PEM agent can be used, you must register it with a PEM server. **PEM agents installed by the PEM server package are registered automatically during server configuration.** For all other agents you must follow these instructions.

### Note

After upgrading the PEM agent, you need to restart it. You don't need to register it again.

## How to register PEM agents

On Linux and Windows hosts, the PEM agent package includes a command line utility called pemworker. You can use it to perform management tasks, including [registering the PEM agent](#).

On Windows, the PEM agent graphical installer allows you to register the agent when installing it. This convenience option doesn't support all the possibilities provided by pemworker. If you don't want the installer to register the agent, clear the **Register now** checkbox. For more details, see the [installation instructions](#).

## Registering a PEM agent using the pemworker utility

The pemworker utility is installed automatically with the PEM agent. It's located in the `/usr/edb/pem/agent/bin` directory on Linux and `C:\Program Files\edb\pem\agent-x64\bin` on Windows. To register an agent, invoke the utility and add the relevant options from the table as needed. Follow each option with a corresponding value.

### Linux

```
# Running as root
export PEM_SERVER_PASSWORD=edb
pemworker --register-agent
```

## Windows

```
set PEM_SERVER_PASSWORD=edb
# Running as admin
./pemworker.exe REGISTER
```

| Option                                     | Description   |
|--|---|
| <code>--pem-server</code>                  | The IP address of the PEM backend database server. This parameter is required.  |
| <code>--pem-port</code>                    | The port of the PEM backend database server. The default value is <code>5432</code> .   |
| <code>--pem-user</code>                    | The name of the database user having superuser privileges of the PEM backend database server. This parameter is required.   |
| <code>--pem-agent-user</code>              | The agent user to connect the PEM server backend database server.   |
| <code>--cert-path</code>                   | The complete path to the directory where certificates are created. If you don't provide a path, certificates are created in <code>~/.pem</code> on Linux and <code>%APPDATA%/pem</code> on Windows.                       |
| <code>--config-dir</code>                  | The directory path for the configuration file. The default is <code>&lt;pemworker path&gt;/../etc</code> .  |
| <code>--display-name</code>                | A user-friendly name for the agent to display in the PEM browser tree. The default is the system hostname.  |
| <code>--force-registration</code>          | Include the <code>force_registration</code> clause to register the agent with the arguments provided. This clause is useful if you're overriding an existing agent configuration. The default value is <code>Yes</code> . |
| <code>--group</code>                       | The name of the group in which the agent is displayed.  |
| <code>--team</code>                        | The name of the database role on the PEM backend database server with access to the monitored database server.  |
| <code>--owner</code>                       | The name of the database user on the PEM backend database server who owns the agent.  |
| <code>--allow-server-restart</code>        | Enable the <code>allow-server-restart</code> parameter to allow PEM to restart the monitored server. The default value is <code>True</code> .   |
| <code>--allow-batch-probes</code>          | Enable the <code>allow-batch-probes</code> parameter to allow PEM to run batch probes on this agent. The default value is <code>False</code> .  |
| <code>--batch-script-user</code>           | The operating system user to use for executing the batch/shell scripts. The default value is none. The scripts don't execute if you leave this parameter blank or the specified user doesn't exist.                       |
| <code>--enable-heartbeat-connection</code> | Enable the <code>enable-heartbeat-connection</code> parameter to create a dedicated heartbeat connection between the PEM agent and server to update the active status. The default value is <code>False</code> .          |
| <code>--enable-smtp</code>                 | Enable the <code>enable-smtp parameter</code> to allow the PEM agent to send the email on behalf of the PEM server. The default value is <code>False</code> .   |
| <code>--enable-snmp</code>                 | Enable the <code>enable-snmp parameter</code> to allow the PEM agent to send the SNMP traps on behalf of the PEM server. The default value is <code>False</code> .  |

### Allowing the agent to restart the database server

If you use any feature of PEM that requires a database server restart by the PEM agent (such as Audit Manager, Log Manager, or the Tuning Wizard), then you must set the value of `allow_server_restart` to `true` in the `agent.cfg` file or restart the server manually for changes to take effect.

### Running shell/batch jobs

If you want to run shell/batch jobs using an agent, you must specify the user for the `batch_script_user` parameter. We strongly recommend that you use a non-root user to run the scripts. Using the root user might result in compromising the data security and operating system security.

Authenticating the pemworker utility

Before any changes are made on the PEM database, the connecting is authenticated with the PEM database server. When invoking the pemworker utility, you must provide the password associated with the PEM server administrative user role (postgres). You can specify the administrative password in three ways:

- Set the `PEM_SERVER_PASSWORD` environment variable.
- Provide the password on the command line with the `PGPASSWORD` keyword.
- Create an entry in the `.pgpass` file.

If you don't provide the password, a password authentication error occurs. After authentication succeeds, you're prompted for any other missing required information. When the registration is complete, the server confirms that the agent was successfully registered.

Unregistering a PEM agent

You can use the pemworker utility to unregister a PEM agent. To unregister an agent, invoke the pemworker utility as shown.

Linux

```
# Running as root
pemworker --unregister-agent
```

Windows

```
./pemworker.exe UNREGISTER-AGENT
```

When invoking the pemworker utility, append command line options to the command string. Follow each option with a corresponding value.

| Option                                   | Description  |
|--|--|
| <code>--pem-user &lt;username&gt;</code> | Specifies the name of the database user (member of pem_admin role) of the PEM backend database server. This parameter is required. |
| <code>--config-dir</code>                | Specifies the directory path for the configuration file. The default is " <code>&lt;pemworker path&gt;/../etc</code> ".            |

Advanced usage

These are some advanced options for PEM agent registration.

Using a non-root user account to register a PEM agent on Linux

To use a non-root user account to register a PEM agent, you must first install the PEM agent as a root user. After installation, assume the identity of a non-root user, such as edb. Then:

1. Log in as edb. Create `pem` and `logs` directories and assign read, write, and execute permissions:

```
$ mkdir /home/edb/pem
$ mkdir /home/edb/pem/logs
$ chmod 700 /home/edb/pem
$ chmod 700 /home/edb/pem/logs
```

2. Register the agent with PEM server:

```
$ export PEM_SERVER_PASSWORD=edb

# Use the following command to create agent certificates and an agent
# configuration file (`agent.cfg`) in the `/home/edb/pem` directory.
$ /usr/edb/pem/agent/bin/pemworker --register-agent --pem-server <172.19.11.230> --pem-user postgres --
--pem-port 5432 --display-name non_root_pem_agent --cert-path /home/edb/pem --config-dir /home/edb/pem

# Use the following command to assign read and write permissions to
# these files:
$ chmod -R 600 /home/edb/pem/agent*
```

3. Change the parameters of the `agent.cfg` file:

```
$ vi /home/edb/pem/agent.cfg
agent_ssl_key=/home/edb/pem/agent<id>.key
agent_ssl_crt=/home/edb/pem/agent<id>.crt
log_location=/home/edb/pem/worker.log
agent_log_location=/home/edb/pem/agent.log
```

Where `<id>` is the assigned PEM agent ID.

4. Create a `tmp` directory, set the environment variable, and start the agent:

```
$ mkdir /home/edb/pem/tmp

# Create a script file, add the environment variable, give permissions, and
# execute:
$ vi
/home/edb/pem/run_pemagent.sh
#!/bin/bash
export TEMP=/home/edb/agent/tmp
/usr/edb/pem/agent/bin/pemagent -c
/home/edb/agent/agent.cfg
$ chmod a+x
/home/edb/pem/run_pemagent.sh
$ cd /home/edb/pem
$ ./run_pemagent.sh
```

Your PEM agent is now registered and started with the edb user. If your machine restarts, then this agent doesn't restart automatically. You need to start it manually using the previous command.

5. Optionally, you can create the service for this PEM agent as the root user to start this agent at machine restart as follows:

- a. Update the values for the configuration file path and the user in the `pemagent` service file as superuser:

```
$ sudo vi /usr/lib/systemd/system/pemagent.service
[Service]
Type=forking
WorkingDirectory=/home/edb/pem
Environment=LD_LIBRARY_PATH=/usr/edb/pem/agent/lib:/usr/libexec/edb-snmpp+/lib
Environment=TEMP=/home/edb/pem/tmp
```

```
ExecStart=/usr/edb/pem/agent/bin/pemagent -c
/home/edb/pem/agent.cfg
```

b. Stop the running agent process, and then restart the agent service:

```
# Find the process id of the running pem agent and pem worker process and kill that process
$ ps -ax | grep pemagent
$ kill -9 <process_id_of_pemagent>
$ ps -ax | grep pemworker
$ kill -9 <process_id_of_pemworker>
# Enable and start pemagent service
$ sudo systemctl enable pemagent
$ sudo systemctl start pemagent
$ sudo systemctl status pemagent
```

6. Check the agent status on the PEM dashboard.

#### Note

Any probes and jobs that require root permission or access to a file owned by another user (for example, enterprisedb) fail.

## 15 Managing a PEM server

Some of the tasks related to managing the PEM server include:

- Restarting the PEM server and agent
- Controlling the PEM server or PEM agent
- Controlling the HTTPD service on Linux and Windows
- Controlling the HTTPD server
- Managing PEM authentication and security
- Modifying the `pg_hba.conf` file
- Modifying PEM to use a proxy server

### Starting and stopping the PEM server and agents

The PEM server starts, stops, and restarts when the Postgres server instance on where it resides starts, stops, or restarts. Use the same commands to control the PEM server that you use to control the Postgres server. On Linux platforms, the command that stops and starts the service script varies by platform and OS version.

The PEM agent is controlled by a service named pemagent.

The Windows operating system includes a graphical service controller that displays the server status and offers interactive server control. You can access the Services utility through the Windows Control Panel. When the utility opens, navigate through the listed services and select the service name.

Select **Stop**, **Pause**, **Start**, or **Restart** to control the state of the service.

Any user or client application connected to the Postgres server is abruptly disconnected if you stop the service. For more information about controlling a service, see [EDB Postgres Advanced Server](#).

### Controlling the PEM server or PEM agent on Linux

On Linux platforms, the name of the service script that controls:

- A PEM server on EDB Postgres Advanced Server is `edb-as-<x>` or `ppas-<x>`.
- A PEM server on PostgreSQL is `postgresql-<x>`.
- A PEM agent is `pemagent`.

Where `x` indicates the server version number.

You can use the service script to control the service.

- To control a service on RHEL 8.x, at the command prompt, assume superuser privileges and issue the command:

```
systemctl <service_name> <action>
```

Where:

`service_name` is the name of the service.

`action` specifies the action taken by the service. Specify:

- `start` to start the service.
- `stop` to stop the service.
- `restart` to stop and then start the service.
- `status` to check the status of the service.

## Controlling the PEM server or PEM agent on Windows

The Windows operating system includes a graphical service controller that displays the server status and offers interactive server control. The registered name of the service that controls:

- A PEM server host on PostgreSQL is `postgresql-<x>`.
- A PEM server host on Advanced Server is `edb-as-<x>` or `ppas-<x>`.
- A PEM agent is `Postgres Enterprise Manager - pemAgent`.

Where `x` indicates the server version number.

In the Windows Control Panel, open the Services utility to see the list of services.

Select **Stop the service** to stop a service. Any user or client application connected to the server is abruptly disconnected if you stop the service.

Select **Pause the service** to reload a service's configuration parameters. **Pause the service** is an effective way to reset parameters without disrupting user sessions for many of the configuration parameters.

Select **Start the service** to start a service.

## Controlling the HTTPD server

On Linux, you can confirm the status of the PEM-HTTPD service by entering the following command at the command prompt:

```
ps -ef | grep httpd
```

If Linux responds with an answer like the following, httpd isn't running:

```
user 13321 13267 0 07:37 pts/1 00:00:00 grep httpd
```

To start the service on a RHEL 8.x system, use the command:

```
systemctl start httpd
```

On Windows, you can use the Services applet to check the status of the PEM HTTPD service. After opening the Services applet, select the PEM HTTPD service from the list.

The Status column displays the current state of the server. Select **Start** to start PEM HTTPD if the service isn't running.

## Managing PEM authentication

Postgres supports a number of authentication methods:

- Secure password (md5)
- GSSAPI
- SSPI
- Kerberos
- Ident
- LDAP
- RADIUS
- Certificate (SSL)
- PAM

Postgres and PEM authentication is controlled by the `pg_hba.conf` configuration file. Entries in the configuration file specify:

- Who can connect to a specific database
- The type of authentication required before that user is allowed to connect

A typical entry in the `pg_hba.conf` file that allows a user named postgres to connect to all databases from the local host (127.0.0.1/32) using secure password (md5) authentication connections takes the form:

```
host all postgres 127.0.0.1/32 md5
```

Depending on your system configuration, you might also need to create a password file for the user account that the PEM agent uses to connect to the server. This file allows the agent to properly respond to the server's authentication request. An entry in the password file for a user named postgres, with a password of `1safepwd` takes the form:

```
localhost:5432:*.postgres:1safepwd
```

The password file is usually named `~root/.pgpass` on Linux systems or `%APPDATA%\postgresql\pgpass.conf` on Windows. For more information about configuring a password file, see the [PostgreSQL website](#).

For more information about the authentication methods supported by Postgres, see the [PostgreSQL core documentation](#).

## Editing the PEM server configuration

You can use the PEM client to manage the configuration parameters of the PEM server to enable features or modify default settings. To open the Server Configuration dialog box, select **Management > Server Configuration**. Then edit the parameter values as needed.

## Managing security

PEM provides an interface for managing your Postgres roles and servers.

### Login roles

When you connect to the PEM server, you must provide role credentials that allow access to the database on which the PEM server stores data. By default, the postgres superuser account is used to initially connect to the server, but it is strongly recommended (for both security and auditing purposes) that individual roles are created for each connecting user. You can use the PEM Query tool, the PEM web interface **Create – Login/Group Role** dialog, or a command line client (such as psql) to create a role.

To use the Create – Login/Group Role dialog box to create a role:

1. In the PEM tree, expand the node for the server where you want to create the role.
2. Right-click the **Login/Group Roles** node.
3. From the context menu, select **Create > Login/Group Role**.
4. Use the tabs of the Create – Login/Group Role dialog box to define the role.
5. After you finish defining the new role, select **Save** to create the role.

To modify the properties of an existing login role, right-click the name of a login role in the tree and select **Properties** from the context menu. To delete a login role, right-click the name of the role and select **Delete/Drop** from the context menu.

For more complete information about creating and managing a role, see the [PostgreSQL online documentation](#).

### Group roles

Group roles can serve as containers and are used to dispense system privileges, such as creating databases, and object privileges, such as inserting data into a table. The primary purpose of a group role is to make the mass management of system and object permissions easier for a DBA. Rather than assigning or modifying privileges individually across many different login accounts, you can assign or change privileges for a single role and then grant that role to many login roles at once.

Use the **Group Roles** node (located beneath the name of each registered server in the PEM tree) to create and manage group roles. Options on the context menu provide access to a dialog box that allows you to create a new role or modify the properties of an existing role. You can find more information about creating roles in the [PostgreSQL documentation](#).

### Using PEM predefined roles to manage access to PEM functionality

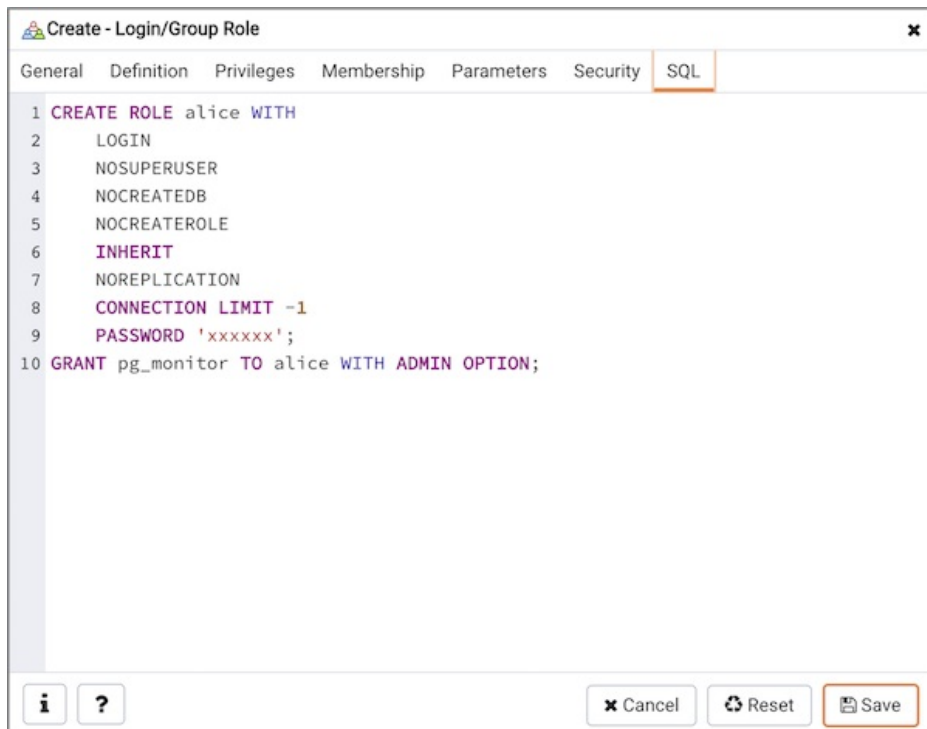
You can use the Login/Group Role dialog box to allow a role with limited privileges to access PEM features such as the Audit Manager, Capacity Manager, or SQL Profiler. PEM predefined roles allow access to PEM functionality. Roles that are assigned membership in these roles can access the associated feature.

When defining a user, use the **Membership** tab to specify the roles in which the new user is a member. The new user shares the privileges associated with each role in which it is a member. For a user to have access to PEM extended functionality, the role must be a member of the pem\_user role and the predefined role that grants access to the feature. Use the **Roles** field to select predefined role names from a list.

The **SQL** tab displays the SQL command that the server executes when you select **Save**.

This example creates a login role named acctg\_clerk that has access to the Audit Manager. The role can make unlimited connections to the server at any given time.





The image shows a 'Create - Login/Group Role' dialog box with tabs for General, Definition, Privileges, Membership, Parameters, Security, and SQL. The SQL tab is active, displaying the following SQL code:

```

1 CREATE ROLE alice WITH
2     LOGIN
3     NOSUPERUSER
4     NOCREATEDB
5     NOCREATEROLE
6     INHERIT
7     NOREPLICATION
8     CONNECTION LIMIT -1
9     PASSWORD 'xxxxxx';
10 GRANT pg_monitor TO alice WITH ADMIN OPTION;

```

At the bottom of the dialog are buttons for 'Cancel', 'Reset', and 'Save'.

You can use PEM predefined roles to allow access to the functionality listed in the table.

| Value                            | Parent role                     | Description  |
|----------------------------------|---------------------------------|--|
| pem_super_admin                  |                                 | Role to manage/configure everything on PEM   |
| pem_admin                        | pem_super_admin                 | Role for administration/management/configuration of all visible agents/servers and monitored objects                             |
| pem_config                       | pem_admin                       | Role for configuration management of PEM   |
| pem_component                    | pem_admin                       | Role to run/execute all wizard/dialog box-based components   |
| pem_rest_api                     | pem_admin                       | Role to access the REST API  |
| pem_server_service_manager       | pem_admin                       | Role for allowing to restart/reload the monitored database server (if server-id provided)  |
| pem_manage_schedule_task         | pem_admin                       | Role to configure the schedule tasks   |
| pem_manage_alert                 | pem_admin                       | Role for managing/configuring alerts and its templates   |
| pem_config_alert                 | pem_config,<br>pem_manage_alert | Role for configuring the alerts on any monitored objects   |
| pem_manage_probe                 | pem_admin                       | Role to create, update, and delete the custom probes and change custom probe configuration                                       |
| pem_config_probe                 | pem_config,<br>pem_manage_probe | Role for probe configuration (history retention, execution frequency, enable/disable the probe) on all visible monitored objects |
| pem_database_server_registration | pem_admin                       | Role to register a database server   |
| pem_comp_postgres_expert         | pem_component                   | Role to run the Postgres Expert  |
| pem_comp_auto_discovery          | pem_component                   | Role to run the auto discovery of a database server dialog box   |
| pem_comp_log_analysis_expert     | pem_component                   | Role to run the Log Analysis Expert  |
| pem_comp_sqlprofiler             | pem_component                   | Role to run the SQL Profiler   |
| pem_manage_efm                   | pem_admin                       | Role to manage Failover Manager  |
| pem_comp_capacity_manager        | pem_component                   | Role to run the Capacity Manager   |
| pem_comp_log_manager             | pem_component                   | Role to run the Log Manager  |
| pem_comp_audit_manager           | pem_component                   | Role to run the Audit Manager  |
| pem_comp_tuning_wizard           | pem_component                   | Role to run the Tuning Wizard  |

## Using a team role

When you register a server for monitoring by PEM, you can specify a team to associate with the server. A team is a group role that you can use to allow or restrict access to one or more monitored servers to a limited group of role members. The PEM client displays a server with a specified team only to those users who are:

- A member of the team role
- The role that created the server
- A role with superuser privileges on the PEM server

To create a team role:

1. In the PEM tree, expand the node for the server where you want to create the role.
2. Right-click the **Login/Group Roles** node.
3. From the context menu, select **Create > Login/Group Role**.
4. In the Create - Login/Group Role dialog box, use the fields to specify the properties of the team role.

## Object permissions

A role must be granted sufficient privileges before accessing, executing, or creating any database object. PEM allows you to assign (**GRANT**) and remove (**REVOKE**) object permissions to group roles or login accounts using the PEM client interface.

Object permissions are managed with the graphical object editor for each particular object. For example, to assign privileges to access a database table, right-click the table name in the tree and select **Properties** from the context menu. Use the **Privileges** tab to assign privileges for the table.

The PEM client also contains a Grant wizard (accessed through the **Tools** menu) that allows you to manage many object permissions at once.

## Server configuration parameters - reference

You can use global configuration options to modify aspects of the PEM Server's behavior. The list of configuration parameters is subject to change.

| Parameter name             | Value and unit | Description  |
|----------------------------|----------------|--|
| audit_log_retention_time   | 30 days        | Specifies the number of days for an audit log to be retained on the PEM server.  |
| auto_create_agent_alerts   | true           | Specifies whether to create default agent level alerts automatically when an agent is registered.  |
| auto_create_server_alerts  | true           | Specifies whether to create default server level alerts automatically when a server is bound to an agent.  |
| chart_disable_bullets      | false          | Enables/disables bullets on line charts on dashboards and Capacity Manager reports.  |
| cm_data_points_per_report  | 50             | Specifies the number of data points to plot on charts on Capacity Manager reports.   |
| cm_max_end_date_in_years   | 5 years        | Specifies the maximum amount of time for the Capacity Manager to extrapolate data for. Ensures that threshold-based end dates of reports aren't extrapolated indefinitely. |
| dash_alerts_timeout        | 60 seconds     | Specifies the number of seconds after which the components of the Alerts dashboard are refreshed.  |
| dash_db_comrol_span        | 7 days         | Specifies the number of days worth of data to plot on the Commit/Rollback Analysis chart on the Database Analysis and Server Analysis dashboards.                          |
| dash_db_comrol_timeout     | 1800 seconds   | Specifies the number of seconds after which the Commits/Rollbacks line chart is refreshed on the Database Analysis and Server Analysis dashboards.                         |
| dash_db_connovervw_timeout | 300 seconds    | Specifies the number of seconds after which the Connection Overview pie chart is refreshed on the Database Analysis dashboard.   |
| dash_db_eventlag_span      | 7 days         | Specifies the number of days worth of data to plot on the Number of Events Lag chart for slony replication on the Database Analysis dashboard.                             |

| Parameter name                  | Value and unit | Description  |
|---------------------------------|----------------|--|
| dash_db_eventlag_timeout        | 1800 seconds   | Specifies the number of seconds after which the Number of Events Lag line chart for slony replication is refreshed on the Database Analysis dashboard.                         |
| dash_db_hottable_rows           | 25 rows        | Specifies the number of rows to show on the HOT Table Analysis table on the Database Analysis dashboard.   |
| dash_db_hottable_timeout        | 300 seconds    | Specifies the number of seconds after which the Hot Tables table is refreshed on the Database Analysis dashboard.  |
| dash_db_io_span                 | 7 days         | Specifies the number of days worth of data to plot on the Database I/O Analysis chart on the Database Analysis and I/O Analysis dashboards.                                    |
| dash_db_io_timeout              | 1800 seconds   | Specifies the number of seconds after which the Database I/O line chart is refreshed on the Database Analysis and I/O Analysis dashboards.                                     |
| dash_db_rowact_span             | 7 days         | Specifies the number of days worth of data to plot on the Row Activity Analysis chart on the Database Analysis, I/O Analysis, and Server Analysis dashboards.                  |
| dash_db_rowact_timeout          | 1800 seconds   | Specifies the number of seconds after which the Row Activity line chart is refreshed on the Database Analysis, I/O Analysis, and Server Analysis dashboards.                   |
| dash_db_storage_timeout         | 300 seconds    | Specifies the number of seconds after which the Storage bar chart is refreshed on the Database Analysis dashboard.   |
| dash_db_timelag_span            | 7 days         | Specifies the number of days worth of data to plot on the Time Lag chart for Slony replication on the Database Analysis dashboard.   |
| dash_db_timelag_timeout         | 1800 seconds   | Specifies the number of seconds after which the Time Lag line chart for Slony replication is refreshed on the Database Analysis dashboard.                                     |
| dash_db_useract_span            | 7 days         | Specifies the number of days worth of data to plot on the User Activity Analysis chart on the Database Analysis dashboard.   |
| dash_db_useract_timeout         | 1800 seconds   | Specifies the number of seconds after which the User Activity line chart is refreshed on the Database Analysis dashboard.  |
| dash_efm_timeout                | 300 seconds    | Specifies the number of seconds after which the Failover Manager Node Status and Failover Manager Cluster Info line chart is refreshed on the Streaming Replication dashboard. |
| dash_global_overview_timeout    | 30 seconds     | Specifies the number of seconds after which the components of the Global Overview dashboard are refreshed.   |
| dash_header_timeout             | 60 seconds     | Specifies the number of seconds after which the information on the header of all the dashboards are refreshed.   |
| dash_io_chkpt_span              | 7 days         | Specifies the number of days worth of data to plot on the Checkpoints chart on the I/O Analysis dashboard.   |
| dash_io_chkpt_timeout           | 1800 seconds   | Specifies the number of seconds after which the Checkpoints line chart is refreshed on the I/O Analysis dashboard.   |
| dash_io_hotindx_timeout         | 300 seconds    | Specifies the number of seconds after which the Hot Indexes bar chart is refreshed on the I/O Analysis dashboard.  |
| dash_io_hottbl_timeout          | 300 seconds    | Specifies the number of seconds after which the Hot Tables bar chart is refreshed on the I/O Analysis dashboard.   |
| dash_io_index_objectio_rows     | 25 rows        | Specifies the number of rows displayed on the Index Activity table on the I/O Analysis and Object Activity Analysis dashboards.  |
| dash_io_index_objectio_timeout  | 60 seconds     | Specifies the number of seconds after which the Index Activity table is refreshed on the I/O Analysis and Object Activity Analysis dashboards.                                 |
| dash_io_objectio_rows           | 25 rows        | Specifies the number of rows displayed on the Object I/O Details table on the I/O Analysis and Object Activity Analysis dashboards.  |
| dash_io_objectio_timeout        | 300 seconds    | Specifies the number of seconds after which the Object I/O Details table is refreshed on the I/O Analysis and Object Activity Analysis Dashboards.                             |
| dash_memory_hostmemact_span     | 7 days         | Specifies the number of days worth of data to plot on the Host Memory Activity Analysis chart on the Memory Analysis dashboard.  |
| dash_memory_hostmemact_timeout  | 1800 seconds   | Specifies the number of seconds after which the Host Memory Activity line chart is refreshed on the Memory Analysis dashboard.   |
| dash_memory_hostmemconf_timeout | 300 seconds    | Specifies the number of seconds after which the Host Memory Configuration pie chart is refreshed on the Memory Analysis and Server Analysis dashboards.                        |

| Parameter name                       | Value and unit | Description   |
|--------------------------------------|----------------|---|
| dash_memory_servmemact_span          | 7 days         | Specifies the number of days worth of data to plot on the server Memory Activity Analysis chart on the Memory Analysis dashboard.       |
| dash_memory_servmemact_timeout       | 1800 seconds   | Specifies the number of seconds after which the Server Memory Activity line chart is refreshed on the Memory Analysis dashboard.        |
| dash_memory_servmemconf_timeout      | 300 seconds    | Specifies the number of seconds after which the Server Memory Configuration pie chart is refreshed on the Memory Analysis dashboard.    |
| dash_objectact_objstorage_rows       | 15 rows        | Specifies the number of rows to show on the Object Storage table on the Object Activity Analysis dashboard.                             |
| dash_objectact_objstorage_timeout    | 300 seconds    | Specifies the number of seconds after which the Object Storage table is refreshed on the Object Activity Analysis dashboard.            |
| dash_objectact_objtopindexes_timeout | 300 seconds    | Specifies the number of seconds after which the Top 5 Largest Indexes bar chart is refreshed on the Object Activity Analysis dashboard. |
| dash_objectact_objtoptables_timeout  | 300 seconds    | Specifies the number of seconds after which the Top 5 Largest Tables bar chart is refreshed on the Object Activity Analysis dashboard.  |
| dash_os_cpu_span                     | 7 days         | Specifies the number of days worth of data to plot on the CPU chart on the Operating System Analysis dashboard.                         |
| dash_os_cpu_timeout                  | 1800 seconds   | Specifies the number of seconds after which the CPU line chart is refreshed on the Operating System Analysis dashboard.                 |
| dash_os_data_span                    | 7 days         | Specifies the number of days worth of data to plot on the I/O line chart on the Operating System Analysis dashboard.                    |
| dash_os_disk_span                    | 7 days         | Specifies the number of days worth of data to plot on the Utilisation chart on the Operating System Analysis dashboard.                 |
| dash_os_hostfs_timeout               | 1800 seconds   | Specifies the number of seconds after which the Host File System Details table is refreshed on the Operating System Analysis dashboard. |
| dash_os_io_timeout                   | 1800 seconds   | Specifies the number of seconds after which the I/O line chart is refreshed on the Operating System Analysis dashboard.                 |
| dash_os_memory_span                  | 7 days         | Specifies the number of days worth of data to plot on the Memory chart on the Operating System Analysis dashboard.                      |
| dash_os_memory_timeout               | 1800 seconds   | Specifies the number of seconds after which the Memory line chart is refreshed on the Operating System Analysis dashboard.              |
| dash_os_packet_span                  | 7 days         | Specifies the number of days worth of data to plot on the Packet chart on the Operating System Analysis dashboard.                      |
| dash_os_packet_timeout               | 1800 seconds   | Specifies the number of seconds after which the Network Packets line chart is refreshed on the Operating System Analysis dashboard.     |
| dash_os_process_span                 | 7 days         | Specifies the number of days worth of data to plot on the Process chart on the Operating System Analysis dashboard.                     |
| dash_os_process_timeout              | 1800 seconds   | Specifies the number of seconds after which the Process line chart is refreshed on the Operating System Analysis dashboard.             |
| dash_os_storage_timeout              | 1800 seconds   | Specifies the number of seconds after which the Storage pie chart is refreshed on the Operating System Analysis dashboard.              |
| dash_os_traffic_span                 | 7 days         | Specifies the number of days worth of data to plot on the Traffic chart on the Operating System Analysis dashboard.                     |
| dash_os_traffic_timeout              | 1800 seconds   | Specifies the number of seconds after which the Traffic line chart is refreshed on the Operating System Analysis dashboard.             |
| dash_os_util_timeout                 | 1800 seconds   | Specifies the number of seconds after which the Utilization line chart is refreshed on the Operating System Analysis dashboard.         |
| dash_probe_log_timeout               | 300 seconds    | Specifies the number of seconds after which the Probe Log table refreshed.  |
| dash_replication_archivestat_span    | 7 days         | Specifies the number of days worth of data to plot on the WAL Archive Status chart on the Streaming Replication Analysis dashboard.     |
| dash_replication_archivestat_timeout | 1800 seconds   | Specifies the number of seconds after which the WAL Archive Status line chart is refreshed on the Streaming Replication dashboard.      |
| dash_replication_pagelag_span        | 7 days         | Specifies the number of days worth of data to plot on the WAL Lag Pages chart on the Streaming Replication dashboard.                   |

| Parameter name                      | Value and unit | Description  |
|-------------------------------------|----------------|--|
| dash_replication_pagelag_timeout    | 1800 seconds   | Specifies the number of seconds after which the WAL Lag Pages line chart is refreshed on the Streaming Replication dashboard.                |
| dash_replication_segmentlag_span    | 7 days         | Specifies the number of days worth of data to plot on the WAL Lag Segments chart on the Streaming Replication dashboard.                     |
| dash_replication_segmentlag_timeout | 1800 seconds   | Specifies the number of seconds after which the WAL Lag Segments line chart is refreshed on the Streaming Replication dashboard.             |
| dash_replication_timelag_span       | 7 days         | Specifies the number of days worth of data to plot on the Replication Lag Time chart on the Streaming Replication dashboard.                 |
| dash_replication_timelag_timeout    | 1800 seconds   | Specifies the number of seconds after which the Replication Lag Time line chart is refreshed on the Streaming Replication dashboard.         |
| dash_server_buffers_written         | 168 hours      | Specifies the number of days worth of data to plot on the Background Writer Statistics chart on the Server Analysis dashboard.               |
| dash_server_buffers_written_timeout | 300 seconds    | Specifies the number of seconds after which the Background Writer Statistics line chart is refreshed on the Server Analysis dashboard.       |
| dash_server_connovervw_timeout      | 300 seconds    | Specifies the number of seconds after which the Connection Overview pie chart is refreshed on the Server Analysis dashboard.                 |
| dash_server_database_timeout        | 300 seconds    | Specifies the number of seconds after which the Databases table is refreshed on the Server Analysis dashboard.                               |
| dash_server_dbsize_span             | 7 days         | Specifies the number of days worth of data to plot on the Database Size Analysis on the Server Analysis dashboard.                           |
| dash_server_dbsize_timeout          | 1800 seconds   | Specifies the number of seconds after which the Database Size line chart is refreshed on the Server Analysis dashboard.                      |
| dash_server_disk_timeout            | 1800 seconds   | Specifies the number of seconds after which the Disk line chart is refreshed on the Server Analysis dashboard.                               |
| dash_server_global_span             | 7 days         | Specifies the number of days worth of data to plot on the Disk line chart on the Server Analysis dashboard.                                  |
| dash_server_sharedbuff_span         | 7 days         | Specifies the number of days worth of data to plot on the Shared Buffer chart on the Server Analysis dashboard.                              |
| dash_server_sharedbuff_timeout      | 1800 seconds   | Specifies the number of seconds after which the Shared Buffers line chart is refreshed on the Server Analysis dashboard.                     |
| dash_server_tabspacesize_span       | 7 days         | Specifies the number of days worth of data to plot on the Tablespace Size chart on the Server Analysis dashboard.                            |
| dash_server_tabspacesize_timeout    | 1800 seconds   | Specifies the number of seconds after which the Tablespace Size line chart is refreshed on the Server Analysis dashboard.                    |
| dash_server_useract_span            | 7 days         | Specifies the number of days worth of data to plot on the User Activity chart on the Server Analysis dashboard.                              |
| dash_server_useract_timeout         | 1800 seconds   | Specifies the number of seconds after which the User Activity line chart is refreshed on the Server Analysis dashboard.                      |
| dash_sessact_lockact_timeout        | 300 seconds    | Specifies the number of seconds after which the Session Lock Activity table is refreshed on the Session Activity Analysis dashboard.         |
| dash_sessact_workload_timeout       | 300 seconds    | Specifies the number of seconds after which the Session Workload table is refreshed on the Session Activity Analysis dashboard.              |
| dash_sess_waits_timewait_timeout    | 300 seconds    | Specifies the number of seconds after which the Session Waits By Time Waited pie chart is refreshed on the Session Waits Analysis dashboard. |
| dash_sess_waits_waitdtl_timeout     | 300 seconds    | Specifies the number of seconds after which the Session Waits Details table is refreshed on the Session Waits Analysis dashboard.            |
| dash_storage_dbdtls_timeout         | 300 seconds    | Specifies the number of seconds after which the Database Details table is refreshed on the Storage Analysis dashboard.                       |
| dash_storage_dbovervw_timeout       | 300 seconds    | Specifies the number of seconds after which the Database Overview pie chart is refreshed on the Storage Analysis dashboard.                  |
| dash_storage_hostdtls_timeout       | 300 seconds    | Specifies the number of seconds after which the Host Details table is refreshed.   |
| dash_storage_hostovervw_timeout     | 300 seconds    | Specifies the number of seconds after which the Host Overview pie chart is refreshed on the Storage Analysis dashboard.                      |

| Parameter name                    | Value and unit | Description   |
|-----------------------------------|----------------|---|
| dash_storage_tblspcdtls_timeout   | 300 seconds    | Specifies the number of seconds after which the Tablespace Details table is refreshed on the Storage Analysis dashboard.  |
| dash_storage_tblspcovervw_timeout | 300 seconds    | Specifies the number of seconds after which the Tablespace Overview pie chart is refreshed on the Storage Analysis dashboard.   |
| dash_sys_waits_nowaits_timeout    | 300 seconds    | Specifies the number of seconds after which the System Waits By Number Of Waits pie chart is refreshed on the System Waits Analysis dashboard.  |
| dash_sys_waits_timewait_timeout   | 300 seconds    | Specifies the number of seconds after which the System Waits By Time Waited pie chart is refreshed on the System Waits Analysis dashboard.  |
| dash_sys_waits_waitdtl_timeout    | 300 seconds    | Specifies the number of seconds after which the System Waits Details table is refreshed on the System Waits Analysis dashboard.   |
| deleted_charts_retention_time     | 7 days         | Specifies the number of days that a custom chart (displayed on a user-defined dashboard) is stored.   |
| deleted_probes_retention_time     | 7 days         | Specifies the number of days that a custom probe (displayed on a user-defined dashboard) is stored.   |
| download_chart_format             | jpeg           | Specifies the format in which a downloaded chart is stored. Can be jpeg or png.   |
| flapping_detection_state_change   | 3              | Specifies the number of state changes detected within a specified interval to define a given alert as flapping. Flapping starts when more than <code>N</code> state changes have occurred over $[N + 1 * (\min(\text{probe\_interval}) * 2)]$ minutes and the fine state is not None. The default value of <code>N</code> is 2 or 3, and $\min(\text{probe\_interval})$ is the smallest interval for all the probes used by the alert. Flapping ends when ZERO state changes have occurred over $[2 * N * \min(\text{probe\_interval})]$ minutes. |
| job_retention_time                | 30 days        | Specifies the number of days that nonrecurring scheduled tasks and their associated jobs are retained.  |
| long_running_transaction_minutes  | 5 minutes      | Specifies the number of minutes a query executes before being considered long running.  |
| nagios_cmd_file_name              | <file_name>    | Specifies nagios command file to which passive service check results are sent.  |
| nagios_enabled                    | t              | Specifies whether alert notification are submitted to nagios.   |
| nagios_medium_alert_as_critical   | f              | Specifies whether a medium level PEM alert is considered critical in nagios.  |
| nagios_spool_retention_time       | 7 days         | Specifies the number of days to retain nagios messages in the spool table before they are discarded.  |
| probe_log_retention_time          | 30 days        | Specifies the number of days that probe log records are retained.   |
| reminder_notification_interval    | 24 hours       | Specifies the number of hours after which a reminder email is sent in case an alert wasn't cleared.   |
| server_log_retention_time         | 30 days        | Specifies the number of days that the server log is retained on the PEM server.   |
| show_data_tab_on_graph            | false          | If <code>true</code> , a <b>Data</b> tab is added to each graph. Select the <b>Data</b> tab to review the data that's plotted on the graph.   |
| smtp_authentication               | false          | Specifies whether to enable/disable authentication over SMTP.   |
| smtp_enabled                      | true           | Specifies whether to enable/disable sending of emails.  |
| smtp_encryption                   | false          | Specifies whether to send SMTP email using an encrypted connection.   |
| smtp_password                     |                | Specifies the password to use to connect to the SMTP server.  |
| smtp_port                         | 25             | Specifies the SMTP server port to use for sending email.  |
| smtp_server                       | 127.0.0.1      | Specifies the SMTP server host address to use for sending email.  |
| smtp_spool_retention_time         | 7 days         | Specifies the number of days to retain sent email messages in the spool table before they are discarded.  |
| smtp_username                     |                | Specifies the username to used to connect to an SMTP server.  |
| snmp_community                    | public         | Specifies the SNMP community used when sending traps. Used only with SNMPv1 and SNMPv2.   |
| snmp_enabled                      | true           | Specifies whether to enable/disable sending SNMP traps.   |
| snmp_port                         | 162            | Specifies the SNMP server port to use for sending SNMP traps.   |
| snmp_server                       | 127.0.0.1      | Specifies the SNMP server host address to use for sending SNMP traps.   |

| Parameter name               | Value and unit           | Description   |
|------------------------------|--------------------------|---|
| snmp_spool_retention_time    | 7 days                   | Specifies the number of days to retain sent traps in the spool table before they are discarded.   |
| snmp_security_name           |                          | Specifies the user name or security name for sending SNMP traps. Used only with SNMPv3.   |
| snmp_security_engine_id      |                          | Specifies the engine id of the SNMP agent on the SNMP server. Used only with SNMPv3.  |
| snmp_security_level          | NOAUTH_NOPRIV            | Specifies security level. Its possible values are: AUTH_NOPRIV - Authentication, No Privacy or AUTH_PRIV - Authentication, Privacy or NOAUTH_NOPRIV - no Authentication, no Privacy. Used only with SNMPv3. |
| snmp_context_name            |                          | Specifies the context name, the identifier for MIB objects when sending SNMP traps. Used only with SNMPv3.  |
| snmp_context_engine_id       |                          | Specifies the context engine id, the identifier for MIB objects when sending SNMP traps. If not specified, snmp_security_engine_id is used. Used only with SNMPv3.  |
| snmp_authentication_protocol | NONE                     | Specifies the authentication type for SNMP traps. Its possible values are NONE, HMACMD5, and HMACSHA. Used only with SNMPv3.  |
| snmp_privacy_protocol        | NONE                     | Specifies the privacy protocol for SNMP traps. Its possible values are NONE, DES, AES128, IDEA, AES192, or AES256. Used only with SNMPv3.   |
| snmp_authentication_password |                          | Specifies the authentication password associated with security name mentioned in snmp_security_name. Used only for SNMPv3.  |
| snmp_privacy_password        |                          | Specifies the privacy password associated with security name mentioned in snmp_security_name. Used only for SNMPv3.   |
| webclient_help_pg            | EDB hosted documentation | Specifies the location of the online PostgreSQL core documentation.   |

## 16 Managing a PEM agent

### Managing job notifications

In the PEM console, you can configure the settings for sending the SMTP trap on success or failure of a system-generated job listed under scheduled tasks or a custom agent job. You can configure these email notification settings at the following three levels to send email notifications to the specified user group. These levels are shown in order of precedence.

- Job level
- Agent level
- PEM server level (default)

#### Configuring job notifications at job level

You can configure email notification settings at job level only for a custom agent job in one of the following ways:

- For a new agent job, you can configure the email notification settings in the **Notification** tab of the Create Agent Job wizard while creating the job.
- For an existing custom job, you can edit the properties of the job and configure the notification settings.

Use the **Notifications** tab to configure the email notification settings on job level:

- Use the **Send the notifications** field to specify when you want to send the email notifications.
- Use the **Email group** field to specify the email group to send the email notification to.

#### Configuring job notifications at agent level



Select the agent in the tree view, right-click, and select **Properties**. In the Properties dialog box, select the **Job Notifications** tab.

Use the Job notifications tab to configure the email notification settings at agent level:

- Use the **Override default configuration?** switch to specify if you want the agent level job notification settings to override the default job notification settings. Select **Yes** to enable the rest of the settings on this dialog box to define when and to whom to send the job notifications.
- Use the **Email on job completion?** switch to specify whether to send the job notification when the job completes successfully.
- Use the **Email on a job failure?** switch to specify whether to send the job notification when the job fails.
- Use the **Email group** field to specify the email group to send the job notification to.

### Configuring job notifications at server level

You can use the Server Configuration dialog box to provide information about your email notification configuration at PEM server level. To open the Server Configuration dialog box, select **Management > Server Configuration**.

Four server configuration parameters specify information about your job notification preferences at PEM server level:

- Use the **job\_failure\_notification** switch to specify if you want to send an email notification after each job failure.
- Use the **job\_notification\_email\_group** parameter to specify the email group to send the email notification to.
- Use the **job\_retention\_time** parameter to specify the number of days to retain nonrecurring scheduled tasks in the system.
- Use the **job\_status\_change\_notification** switch to specify if you want to send an email notification after each job status change, that is, failure, success, or interrupted.

### Managing PEM scheduled jobs

You can create a PEM scheduled job to perform a set of steps you define in the specified sequence. These steps can contain SQL code or a batch/shell script that you can run on a server that's bound with the agent. You can schedule these jobs to suit your business requirements. For example, you can create a job for taking a backup of a particular database server and schedule it to run on a specific date and time of every month.

To create or manage a PEM scheduled job, use the PEM tree to browse to the PEM agent for which you want to create the job. The tree displays a **Jobs** node, under which currently defined jobs are listed. To add a job, right-click the **Jobs** node and select **Create Job** from the context menu.

Use the tabs on the Create - Agent Job dialog box to define the steps and schedule that make up a PEM scheduled job.

Use the **General** tab to provide general information about a job:

- Provide a name for the job in the **Name** field.
- Set the **Enabled** switch to **Yes** to enable a job. Set it to **No** to disable a job.
- Use the **Comment** field to store notes about the job.

Use the **Steps** tab to define and manage the steps that the job performs. Select **Add (+)** to add a step. Then, select the compose icon, located at the left side of the header, to open the Step Definition dialog box.

Use the Step Definition dialog box to define the step:

- Provide a name for the step in the **Name** field. Steps are performed in alphanumeric order by name.
- Use the **Enabled** switch to include the step when executing the job (**True**) or to disable the step (**False**).
- Use the **Kind** switch to indicate if the job step invokes SQL code (**SQL**) or a batch script (**Batch**).
  - If you select **SQL**, use the **Code** tab to provide SQL code for the step.
  - If you select **Batch**, use the **Code** tab to provide the batch script to execute during the step.



- Use the **On error** list to specify the behavior of pgAgent if it encounters an error while executing the step. Select from:
  - **Fail** — Stop the job if you encounter an error while processing this step.
  - **Success** — Mark the step as completing successfully and continue.
  - **Ignore** — Ignore the error and continue.
- If you selected SQL as your input for the **Kind** switch:
  - Use the **Server** field to specify the server that's bound with the agent for which you are creating the PEM scheduled job.
  - Use the **Database** field to specify the database that's associated with the server that you selected.
- Use the **Comment** field to provide a comment about the step.
- Use the context-sensitive field on the Step Definition dialog box **Code** tab to provide the SQL code or batch script to execute during the step:
  - If the step invokes SQL code, provide one or more SQL statements in the **SQL query** field.
  - If the step invokes a batch script, provide the script in the **Code** field. If you're running on a Windows server, use standard batch file syntax. On a Linux server, you can use any shell script, provided that you specify a suitable interpreter on the first line (such as `#!/bin/sh`). Along with the defined inline code, you can also provide the path of any batch script, shell script, or SQL file on the filesystem.
    - To invoke a script on a Linux system, you must modify the entry for the `batch_script_user` parameter in the `agent.cfg` file and specify the user who runs the script. You can specify either a nonroot user or root for this parameter. If you don't specify a user or the specified user doesn't exist, then the script doesn't execute. Restart the agent after modifying the file.
    - To invoke a script on a Windows system, set the registry entry for `AllowBatchJobSteps` to `true` and restart the PEM agent. PEM registry entries are located in `HKEY_LOCAL_MACHINE\Software\EnterpriseDB\PEM\agent`.

After providing all the information required by the step, select **Save**.

Select **Add (+)** to add each step, or select the **Schedules** tab to define the job schedule.

On the **Schedules** tab, select **Add (+)** to add a schedule for the job. Then select the compose icon located at the left side of the header to open the Schedule Definition dialog box.

Use the **Schedules definition** tab to specify the days and times for the job to execute.

- Provide a name for the schedule in the **Name** field.
- Use the **Enabled** switch to indicate for pgAgent to use the schedule (**Yes**) or to disable the schedule (**No**).
- Use the calendar selector in the **Start** field to specify the starting date and time for the schedule.
- Use the calendar selector in the **End** field to specify the ending date and time for the schedule.
- Use the **Comment** field to provide a comment about the schedule.

Select the **Repeat** tab to define the days when the schedule executes in a cron-style format. The job executes on each date or time element selected on the **Repeat** tab.

In each field, select a value to add it to the list of selected values for the field. To clear the values from a field, select the X located at the right-side of the field.

- Use the fields in the **Days** box to specify the days when the job executes:
  - Use the **Week Days** field to select the days when the job executes.
  - Use the **Month Days** field to select the numeric days when the job executes. Select **Last Day** to perform the job on the last day of the month, regardless of the date.
  - Use the **Months** field to select the months when the job executes.
- Use the fields in the **Times** box to specify the times when the job executes in hours and minutes.

Select the **Exceptions** tab to specify any days when you don't want the schedule to execute. For example, you might not want to run jobs on holidays.

Select **Add (+)** to add a row to the exception table. Then:

- In the **Date** column, open a calendar selector and select a date when you don't want the job to execute. Specify **\*\*<Any>\*\*** to indicate that you don't want the job to execute on any day at the time selected.
- In the **Time** column, open a time selector and specify a time when you don't want the job to execute. Specify **\*\*<Any>\*\*** to indicate that you don't want the job to execute at any time on the day selected.

Select the **Notifications** tab to configure the email notification settings on job level:

- Use the **Send the notifications** field to specify when you want to send the email notifications.
- Use the **Email group** field to specify the email group to send the email notification to.

When you finish defining the schedule, you can use the **SQL** tab to review the code that creates or modifies your job.

Select **Save** to save the job definition.

After you save a job, the job is listed under the **Jobs** node of the PEM tree of the server on which it was defined. The **Properties** tab in the PEM console displays a high-level overview of the selected job, and the **Statistics** tab shows the details of each run of the job. To modify an existing job or to review detailed information about a job, right-click a job name, and select **Properties** from the context menu.

Agent privileges

By default, the PEM agent is installed with root privileges for the operating system host and superuser privileges for the database server. These privileges allow the PEM agent to invoke unrestricted probes on the monitored host and database server about system usage, retrieving and returning the information to the PEM server.

PEM functionality lessens as the privileges of the PEM agent decrease. For complete functionality, run the PEM agent as root. If the PEM agent is run under the database server's service account, PEM probes don't have complete access to the statistical information used to generate reports, and functionality is limited to the capabilities of that account. If the PEM agent is run under another lesser-privileged account, functionality is limited even further.

If you limit the operating system privileges of the PEM agent, some of the PEM probes don't return information, and the following functionality might be affected.

Note

The list isn't comprehensive but provides an overview of the type of functionality that's limited.

| Probe or action                                 | Operating system  | PEM functionality affected   |
|---|-------------------|--|
| Data And Logfile Analysis                       | Linux/<br>Windows | The Postgres Expert can't access complete information.   |
| Session Information                             | Linux             | The per-process statistics are incomplete.   |
| PG HBA  | Linux/<br>Windows | The Postgres Expert can't access complete information.   |
| Service restart functionality                   | Linux/<br>Windows | The Audit Log Manager, Server Log Manager Log Analysis Expert, and PEM might not be able to apply requested modifications. |
| Package Deployment                              | Linux/<br>Windows | PEM can't run downloaded installation modules.   |
| Batch Task                                      | Windows           | PEM can't run scheduled batch jobs in Windows.   |
| Collect data from server (root access required) | Linux/<br>Windows | Columns such as swap usage, CPU usage, IO read, IO write appear as 0 in the session activity dashboard.                    |

If you restrict the database privileges of the PEM agent, the following PEM functionality might be affected:

| Probe                      | Operating system | PEM functionality affected   |
|----------------------------|------------------|--|
| Audit Log Collection       | Linux/Windows    | PEM receives empty data from the PEM database.                                     |
| Server Log Collection      | Linux/Windows    | PEM can't collect server log information.  |
| Database Statistics        | Linux/Windows    | The Database/Server Analysis dashboards contain incomplete information.            |
| Session Waits/System Waits | Linux/Windows    | The Session/System Waits dashboards contain incomplete information.                |
| Locks Information          | Linux/Windows    | The Database/Server Analysis dashboards contain incomplete information.            |
| Streaming Replication      | Linux/Windows    | The Streaming Replication dashboard doesn't display information.                   |
| Slony Replication          | Linux/Windows    | Slony-related charts on the Database Analysis dashboard don't display information. |
| Tablespace Size            | Linux/Windows    | The Server Analysis dashboard doesn't display complete information.                |
| xDB Replication            | Linux/Windows    | PEM can't send xDB alerts and traps.   |

If the probe is querying the operating system without enough privileges, the probe might return a `permission denied` error. If the probe is querying the database without enough privileges, the probe might return a `permission denied` error or display the returned data in a PEM chart or graph as an empty value.

When a probe fails, an entry is written to the log file that contains the name of the probe, the reason the probe failed, and a hint that helps you resolve the problem.

You can view probe-related errors that occurred on the server in the Probe Log dashboard or review error messages in the PEM worker log files. On Linux, the default location of the log file is:

```
/var/log/pem/worker.log
```

On Windows, log information is available on the Event Viewer.

## Agent configuration

A number of configurable parameters control the behavior of the PEM agent. On Linux, these are stored in a configuration file. On Windows they are stored as registry keys.

Most agent configuration is managed automatically. We recommend against manually modifying any of these configuration parameters unless specifically directed to do so.

### Note

If you add, remove, or modify any of the parameters, you must restart the agent to apply the changes .

On Linux systems, PEM configuration options are stored in the `agent.cfg` file, located in `/usr/edb/pem/agent/etc` . The `agent.cfg` file contains the following entries.

| Parameter name             | Description  | Default value   |
|----------------------------|--|---|
| <code>pem_host</code>      | The IP address or hostname of the PEM server.  | 127.0.0.1.  |
| <code>pem_port</code>      | The database server port to which the agent connects to communicate with the PEM server. | Port 5432.  |
| <code>pem_agent</code>     | A unique identifier assigned to the PEM agent.   | The first agent is '1', the second agent is '2', and so on. |
| <code>agent_ssl_key</code> | The complete path to the PEM agent's key file.   | <code>/root/.pem/agent.key</code>                           |
| <code>agent_ssl_crt</code> | The complete path to the PEM agent's certificate file.                                   | <code>/root/.pem/agent.crt</code>                           |

| Parameter name            | Description   | Default value   |
|---------------------------|---|---|
| agent_flag_dir            | Used for HA support. Specifies the directory path checked for requests to take over monitoring another server. Requests are made in the form of a file in the specified flag directory.   | Not set by default.   |
| log_level                 | Log level specifies the type of event to write to the PEM log files, one of <code>debug2</code> , <code>debug</code> , <code>info</code> , <code>warning</code> , <code>error</code> . These are in descending order of logging verbosity; <code>debug2</code> logs everything possible, and <code>error</code> only logs errors.           | warning   |
| log_location              | Specifies the location of the PEM worker log file.  | 127.0.0.1.  |
| agent_log_location        | Specifies the location of the PEM agent log file.   | /var/log/pem/agent.log  |
| long_wait                 | The maximum length of time (in seconds) for the PEM agent to wait before attempting to connect to the PEM server if an initial connection attempt fails.  | 30 seconds  |
| short_wait                | The minimum length of time (in seconds) for the PEM agent to wait before checking which probes are next in the queue waiting to run.  | 10 seconds  |
| alert_threads             | The number of alert threads to be spawned by the agent. For more information, see <a href="#">About alert threads</a> .   | Set to 1 for the agent that resides on the host of the PEM server, 0 for all other agents.                    |
| enable_smtp               | When set to true this agent will attempt to send email notifications as configured in the PEM web application.  | true for PEM server host, false for all others.   |
| enable_snmp               | When set to true this agent will attempt to send SNMP notifications as configured in the PEM web application.   | true for PEM server host, false for all others.   |
| enable_nagios             | When set to true, Nagios alerting is enabled.   | true for PEM server host, false for all others.   |
| enable_webhook            | When set to true, Webhook alerting is enabled.  | true for PEM server host, false for all others.   |
| max_webhook_retries       | Set maximum number of times pemAgent retries to call webhooks on failure.   | Default 3.  |
| connect_timeout           | The max time in seconds (a decimal integer string) for the agent to wait for a connection.  | Not set by default. Set to 0 to indicate for the agent to wait indefinitely.                                  |
| allow_server_restart      | If set to TRUE, the agent can restart the database server that it monitors. Some PEM features might be enabled/disabled, depending on the value of this parameter.  | False   |
| max_connections           | The maximum number of probe connections used by the connection throttler.   | 0 (an unlimited number)   |
| connection_lifetime       | Use ConnectionLifetime (or connection_lifetime) to specify the minimum number of seconds an open but idle connection is retained. This parameter is ignored if the value specified in MaxConnections is reached and a new connection to a different database is required to satisfy a waiting request.                                      | By default, set to 0 (a connection is dropped when the connection is idle after the agent's processing loop). |
| allow_batch_probes        | If set to TRUE, the user can create batch probes using the custom probes feature.   | false   |
| heartbeat_connection      | When set to TRUE, a dedicated connection is used for sending the heartbeats.  | false   |
| batch_script_dir          | Provide the path where script file (for alerting) is stored.  | /tmp  |
| connection_custom_setup   | Use to provide SQL code to invoke when a new connection with a monitored server is made.  | Not set by default.   |
| ca_file                   | The path to a CA certificate to be used instead of the platform default for verifying webhook server certificates. When defining webhooks, you can override this behavior with the <code>--webhook_ssl_ca_cert</code> option.   | Not set by default.   |
| batch_script_user         | Provide the name of the user to use for executing the batch/shell scripts.  | None  |
| <b>Webhook parameters</b> | You can specify the following options multiple times. Each time, precede the option with a header of the form <code>[WEBHOOK/&lt;name&gt;]</code> , where <code>&lt;name&gt;</code> is the name of a previously created webhook. These settings are automatically added when webhooks are created. We don't recommend adding them manually. |   |
| webhook_ssl_key           | The complete path to the webhook's SSL client key file.   |   |
| webhook_ssl_cert          | The complete path to the webhook's SSL client certificate file.   |   |
| webhook_ssl_crl           | The complete path of the CRL file to validate webhook server certificate.   |   |
| webhook_ssl_ca_cert       | The complete path to the webhook's SSL ca certificate file.   |   |
| allow_insecure_webhooks   | When set to true, allow webhooks to call with insecure flag.  | false   |

On Windows systems, PEM registry entries are located in:

```
HKEY_LOCAL_MACHINE\Software\EnterpriseDB\PEM\agent
```

The registry contains the entries shown in the table.

| Parameter name        | Description   | Default value   |
|-----------------------|---|---|
| PEM_HOST              | The IP address or hostname of the PEM server.   | 127.0.0.1.  |
| PEM_PORT              | The database server port to which the agent connects to communicate with the PEM server.  | Port 5432.  |
| AgentID               | A unique identifier assigned to the PEM agent.  | The first agent is '1', the second agent is '2', and so on.   |
| AgentKeyPath          | The complete path to the PEM agent's key file.  | %APPDATA%\Roaming\pem\agent.key.  |
| AgentCrtPath          | The complete path to the PEM agent's certificate file.  | %APPDATA%\Roaming\pem\agent.crt   |
| AgentFlagDir          | Used for HA support. Specifies the directory path checked for requests to take over monitoring another server. Requests are made in the form of a file in the specified flag directory.   | Not set by default.   |
| LogLevel              | Specifies the type of event to write to the PEM log files, one of <code>debug2</code> , <code>debug</code> , <code>info</code> , <code>warning</code> , <code>error</code> . These are in descending order of logging verbosity; <code>debug2</code> logs everything possible, and <code>error</code> only logs errors. | warning   |
| LongWait              | The maximum length of time (in seconds) that the PEM agent waits before attempting to connect to the PEM server if an initial connection attempt fails.   | 30 seconds  |
| shortWait             | The minimum length of time in seconds that the PEM agent waits before checking which probes are next in the queue (waiting to run).   | 10 seconds  |
| AlertThreads          | The number of alert threads for the agent to spawn. For more information, see <a href="#">About alert threads</a> .   | Set to 1 for the agent that resides on the host of the PEM server, 0 for all other agents.                    |
| EnableSMTP            | When set to true, the SMTP email feature is enabled.  | true for PEM server host, false for all others.   |
| EnableSNMP            | When set to true, the SNMP trap feature is enabled.   | true for PEM server host, false for all others.   |
| EnableWebhook         | When set to true, Webhook alerting is enabled.  | true for PEM server host, false for all others.   |
| MaxWebhookRetries     | Set maximum number of times for pemAgent to retry to call webhooks on failure.  | Default 3.  |
| ConnectTimeout        | The max time in seconds (a decimal integer string) that the agent waits for a connection.   | Not set by default. If set to 0, the agent waits indefinitely.  |
| AllowServerRestart    | If set to TRUE, the agent can restart the database server that it monitors. Some PEM features might be enabled/disabled, depending on the value of this parameter.  | true  |
| MaxConnections        | The maximum number of probe connections used by the connection throttler.   | 0 (an unlimited number)   |
| ConnectionLifetime    | Use ConnectionLifetime (or connection_lifetime) to specify the minimum number of seconds an open but idle connection is retained. This parameter is ignored if the value specified in MaxConnections is reached and a new connection to a different database is required to satisfy a waiting request.                  | By default, set to 0 (a connection is dropped when the connection is idle after the agent's processing loop). |
| AllowBatchProbes      | If set to TRUE, the user can create batch probes using the custom probes feature.   | false   |
| HeartbeatConnection   | When set to TRUE, a dedicated connection is used for sending the heartbeats.  | false   |
| BatchScriptDir        | Provide the path to store the script file for alerting.   | /tmp  |
| ConnectionCustomSetup | Use to provide SQL code to invoke when a new connection with a monitored server is made.  | Not set by default.   |

| Parameter name            | Description  | Default value |
|---------------------------|--|---------------|
| AllowBatchJobSteps        | If set to true, the batch/shell scripts are executed. Scripts are executed by the user account under which the PEM agent is running.   | None          |
| <b>Webhook parameters</b> | You can specify the following options multiple times. Place them in a subkey <code>Software\EnterpriseDB\PEM\agent\WEBHOOK\&lt;name&gt;</code> , where <code>&lt;name&gt;</code> is the name of a previously created webhook. These settings are automatically added when webhooks are created. We don't recommend adding them manually. |               |
| WebhookSSLKey             | The complete path to the webhook's SSL client key file.  |               |
| WebhookSSLCrt             | The complete path to the webhook's SSL client certificate file.  |               |
| WebhookSSLCrl             | The complete path of the CRL file to validate webhook server certificate.  |               |
| WebhookSSLCaCrt           | The complete path to the webhook's SSL ca certificate file.  |               |
| AllowInsecureWebhooks     | When set to true, allow webhooks to call with insecure flag.   | false         |

### About alert threads

The number of alert threads spawned by an agent is determined by the `alert_threads` or `AlertThreads` parameter. In general it is recommended to set this parameter to 1 on the agent which resides on the PEM server and 0 for all other agents. However on PEM server instances with very large numbers of alerts (caused by many monitored servers, many enabled alerts, or high alert frequency) it may be necessary to increase this parameter if alerts are not being evaluated at the configured frequency. In this situation, it is recommended to set this parameter to around 8 on the agent which resides on the PEM server and 0 for all other agents and tune up or down accordingly.

When tuning this parameter it is important to understand that any agent can process any alert, so in general it is unnecessary to have a non-zero number of alert threads on more than one agent. The capacity of the PEM instance to process alerts is determined by the total number of alert thread across all agents. Increasing the number of threads on a specific agent does not give any additional performance for alerts pertaining to servers monitored by that agent.

Each alert thread opens a connection to the PEM server backend, so allocating more threads than necessary does result in additional memory and CPU usage on the PEM server.

### Agent properties

The PEM Agent Properties dialog box provides information about the PEM agent from which the dialog box was opened. To open the dialog box, right-click an agent name in the PEM client tree and select **Properties** from the context menu.

Use the PEM Agent Properties dialog box to review or modify information about the PEM agent:

- The **Description** field displays a modifiable description of the PEM agent. This description is displayed in the tree of the PEM client.
- You can use groups to organize your servers and agents in the PEM client tree. Use the **Group** list to select the group in which the agent is displayed.
- Use the **Team** field to specify the name of the group role that can access servers monitored by the agent. The servers monitored by this agent are visible in the PEM client tree to connected team members. This is a convenience feature. The **Team** field doesn't provide true isolation. Don't use it for security purposes.
- The **Heartbeat interval** fields display the length of time that elapses between reports from the PEM agent to the PEM server. Use the selectors next to the **Minutes** or **Seconds** fields to modify the interval.

Use the **Job Notifications** tab to configure the email notification settings on agent level:

- Use the **Override default configuration?** switch to specify if you want the agent level job notification settings to override the default job notification settings. Select **Yes** to enable the rest of the settings on this dialog box to define when and to whom to send the job notifications.
- Use the **Email on job completion?** switch to specify whether to send the job notification when the job completes successfully.
- Use the **Email on a job failure?** switch to specify whether to send the job notification when the job fails.

- Use the **Email group** field to specify the email group to send the job notification to.

The **Agent Configurations** tab displays all the current configurations and capabilities of an agent.

- The **Parameter** column displays a list of parameters.
- The **Value** column displays the current value of the corresponding parameter.
- The **Category** column displays the category of the corresponding parameter. It can be either **configuration** or **capability**.

## 17 Managing Postgres servers

### Starting and stopping monitored database servers

PEM lets you start up and shut down managed server instances with the PEM client. To configure a server to allow PEM to manage the service, complete the Server Registration dialog box. Use the dialog box to register the database server with a PEM agent, which is the local agent installed on the same host as the database server. You can also:

- Specify the **Store on PEM Server** option on the Properties dialog box.
- Specify the name of a service script in the **Service ID** field on the **Advanced** tab:
  - For EDB Postgres Advanced Server, the service name is `edb-as-<x>` or `ppas-<x>`.
  - For PostgreSQL, the service name is `postgresql-<x>`.

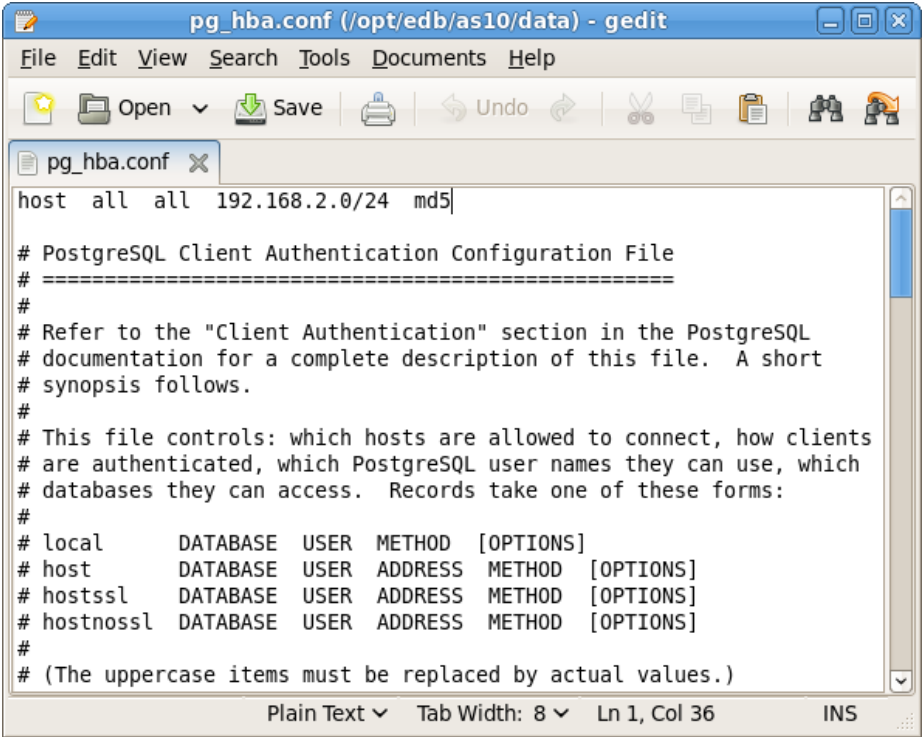
Where `<x>` indicates the server version number.

After connecting to the server, you can start or stop the server by selecting the server name in the tree and selecting **Tools > Queue Server Startup** or **Tools > Queue Server Shutdown**.

### Modifying the `pg_hba.conf` file

Entries in the `pg_hba.conf` file control network authentication and authorization. The `pg_hba.conf` file on the PEM server host must allow connections between the PEM server and PEM-HTTPD, the PEM agent, and the monitored servers.

The PEM server installation process prompts you for the IP address and connection information for hosts for PEM to monitor. This information is added to the top of the `pg_hba.conf` file of the PEM backing database.



You might also need to manually modify the `pg_hba.conf` file to allow connections between the PEM server and other components. For example, if your PEM-HTTPD installation isn't on the same host as the PEM server, you must modify the `pg_hba.conf` file on the PEM server host to allow PEM-HTTPD to connect to the server.

By default, the `pg_hba.conf` file resides in the data directory, under your Postgres installation. For example, on an EDB Postgres Advanced Server 10 host, the default location of `pg_hba.conf` is:

```
/var/lib/edb/as10/data/pg_hba.conf
```

You can modify the `pg_hba.conf` file with your editor of choice. After modifying the file, restart the server for changes to take effect.

The following example shows a `pg_hba.conf` entry that allows an md5 password authenticated connection from a user named postgres to the postgres database on the host where `pg_hba.conf` resides. The connection is coming from an IP address of 192.168.10.102:

| # | TYPE                    | DATABASE | USER     | CIDR-ADDRESS      | METHOD |
|---|-------------------------|----------|----------|-------------------|--------|
| # | IPv4 local connections: |          |          |                   |        |
|   | host                    | postgres | postgres | 192.168.10.102/32 | md5    |

You can specify the address of a network host or a network address range. For example, if you want to allow connections from servers with the addresses 192.168.10.23, 192.168.10.76, and 192.168.10.184, enter a CIDR-ADDRESS of `192.168.10.0/24` to allow connections from all of the hosts in that network:

| # | TYPE                    | DATABASE | USER | CIDR-ADDRESS    | METHOD |
|---|-------------------------|----------|------|-----------------|--------|
| # | IPv4 local connections: |          |      |                 |        |
|   | host                    | postgres | all  | 192.168.10.0/24 | md5    |

For more information about formatting a `pg_hba.conf` file entry, see the [PostgreSQL core documentation](#).

Before you can connect to a Postgres server with PEM, you must ensure that the `pg_hba.conf` file on both servers allows the connection. If you receive an error when connecting to the database server, modify the `pg_hba.conf` file, adding an entry that allows the connection.

Creating and maintaining databases and objects



Each instance of a Postgres server manages one or more databases. Each user must provide authentication information to connect to the database before accessing the information it contains. The PEM client lets you create and manage databases and the objects that comprise a database, such as tables, indexes, and stored procedures.

To create a database in PEM, right-click any managed server's **Databases** node and select **Create > Database**. After defining a database, you can create objects in the new database.

For example, to create a table, right-click a **Tables** node and select **Create > Table**. Specify the attributes of the table in the New Table dialog box.

PEM provides similar dialog boxes for creating and managing other database objects such as:

- Tables
- Indexes
- Stored procedures
- Functions
- Triggers
- Views
- Constraints

Each object type is listed in the tree. Right-click the node that corresponds to an object type to access the **Create** menu and create an object. To perform administrative tasks for the selected object, select **Properties** from the context menu of a named node.

## Template Linux service script

A service script for the database server allows the PEM server to start, stop, or restart the database server. Doing so might be necessary when performing configuration management, certificate management, and other administrative tasks. Service scripts are platform specific.

The Postgres server on which the PEM server resides must contain a service script. Postgres installers in Windows generated by EDB create a service script for you. If you're using a Postgres server from another source like native packages, you must provide a service script.

### Note

On Rocky Linux or RHEL, the service script resides in `/usr/lib/systemd/system`.

For information about customizing a Postgres service, see the [PostgreSQL documentation](#).

## 18 Managing certificates

Files stored in the data directory of the PEM server backing database contain information that helps the PEM server use secure connections:

- `ca_certificate.crt`
- `ca_key.key`
- `server.crt`
- `server.key`
- `root.crl`
- `root.crt`

The PEM agent that's installed with the PEM server monitors the expiration date of the `ca_certificate.crt` file. When the certificate is about to expire, PEM:

- Makes a backup of the existing certificate files.
- Creates new certificate files, appending the new CA certificate file to the `root.crt` file on the PEM server.
- Creates a job that renews the certificate file of any active agents.

- Restarts the PEM server.

When you uninstall an agent, the certificate associated with that agent is added to the certificate revocation list maintained in the `root.crl` file to ensure that you can't use the certificate to connect to the PEM server.

You can manually replace certificate files.

## Replacing SSL certificates

You can replace the SSL certificates on an existing PEM installation. If you plan to upgrade your server to a new version at the same time, invoke all of the PEM installers (first the server installer, then agent installers) before replacing the SSL certificates. Then:

1. Stop all running PEM agents, first on the server host, and then on any monitored node.

To stop a PEM agent on a Linux host, open a terminal window, assume superuser privileges, and enter the command:

On Linux with systemd, for example, Centos 7 or 8

```
systemctl stop pemagent
```

On a Windows host, you can use the Services applet to stop the PEM agent. The PEM agent service is named Postgres Enterprise Manager Agent. Select the service name in the Services dialog box and select **Stop the service**.

2. Take a backup of the existing SSL keys and certificates. The SSL keys and certificates are stored in the `data` directory under your PEM installation. For example, the default location on a Linux system is:

`/var/lib/pgsql/x/data`, where `x` is the PostgreSQL database version

Copy the following files, adding an extension to each file to make the name unique:

- `ca_certificate.crt`
- `ca_key.key`
- `root.crt`
- `root.crl`
- `server.key`
- `server.crt`

For example, to create a backup of the `ca_certificate` file with the word `old` appended to the entry, use this command:

```
# cp ca_certificate.crt ca_certificate_old.crt
```

3. Use the `openssl_rsa_generate_key()` function to generate the `ca_key.key` file:

```
/usr/pgsql-x.x/bin/psql -U postgres -d pem --no-psqlrc -t -A -c "SELECT  
public.openssl_rsa_generate_key(1024)" > /var/lib/pgsql/x/data/ca_key.key
```

After creating the `ca_key.key` file, `cat` the contents to the variable `CA_KEY` for use when generating the `ca_certificate.crt` file. Modify the privileges on the `ca_key.key` file:

```
CA_KEY=$(cat /var/lib/pgsql/x/data/ca_key.key)  
  
chmod 600 /var/lib/pgsql/x/data/ca_key.key
```

4. Use the key to generate the `ca_certificate.crt` file. For simplicity, place the SQL query in a temporary file with a unique name:

```
echo "SELECT openssl_csr_to_crt(openssl_rsa_key_to_csr('${CA_KEY}',
'PEM','US', 'MA', 'Bedford', 'Postgres Enterprise Manager',
'support@enterprisedb.com'), NULL,
'/var/lib/pgsql/x/data/ca_key.key')" > /tmp/_random.$$
```

Then use the variable to execute the query, placing the content in the `ca_certificate.crt` file.

```
/usr/pgsql-x/bin/psql -U postgres -d pem --no-psqlrc -t -A -f /tmp/_random.$$ >
/var/lib/pgsql/x/data/ca_certificate.crt
```

Modify the permissions of the `ca_certificate.crt` file, and remove the temporary file that contained the SQL command:

```
chmod 600 /var/lib/pgsql/x/data/ca_certificate.crt

rm -f /tmp/_random.$$
```

5. Reuse the `ca_certificate.crt` file as the `root.crt` file:

```
cp /var/lib/pgsql/x/data/ca_certificate.crt /var/lib/pgsql/x/data/root.crt
```

Modify the permissions of the `root.crt` file:

```
chmod 600 /var/lib/pgsql/x/data/root.crt
```

6. Use the `openssl_rsa_generate_crl()` function to create the certificate revocation list (`root.crl`):

```
/usr/pgsql-x/bin/psql -U postgres -d pem --no-psqlrc -t -A -c
"SELECT openssl_rsa_generate_crl('/var/lib/pgsql/x/data/ca_certificate.crt',
'/var/lib/pgsql/x/data/ca_key.key')" > /var/lib/pgsql/x/data/root.crl
```

Modify the permissions of the `root.crl` file:

```
chmod 600 /var/lib/pgsql/x/data/root.crl
```

7. Use the `openssl_rsa_generate_key()` function to generate the `server.key` file:

```
/usr/pgsql-x/bin/psql -U postgres -d pem --no-psqlrc -t -A -c "SELECT
public.openssl_rsa_generate_key(1024)" >> /var/lib/pgsql/x/data/server.key
```

After creating the `server.key` file, `cat` the contents to the variable `SSL_KEY` for use when generating the `server.crt` file and modify the privileges on the `server.key` file:

```
SSL_KEY=$(cat /var/lib/pgsql/x/data/server.key)

chmod 600 /var/lib/pgsql/x/data/server.key
```

8. Use the `SSL_KEY` to generate the server certificate. Save the certificate in the `server.crt` file. For simplicity, first place the SQL query into a temporary file with a unique name:

```
echo "SELECT openssl_csr_to_crt(openssl_rsa_key_to_csr('${SSL_KEY}',
```

```
'PEM','US', 'MA', 'Bedford', 'Postgres Enterprise Manager',
'support@enterprisedb.com'),
'/var/lib/pgsql/x/data/ca_certificate.crt',
'/var/lib/pgsql/x/data/ca_key.key')" > /tmp/_random.$$

/usr/pgsql-x/bin/psql -U postgres -d pem --no-psqlrc -t -A -f /tmp/_random.$$ >>
/var/lib/pgsql/x/data/server.crt
```

9. Modify the privileges on the `server.crt` file, and delete the temporary file:

```
chmod 600 /var/lib/pgsql/x/data/server.crt

rm -f /tmp/_random.$$
```

10. Restart the Postgres server:

On Linux with `init.d`, for example, on a Centos6 host:

```
/etc/init.d/postgresql-x restart
```

On Linux with `systemd`, for example, on a Centos7 host:

```
systemctl restart postgresql-x
```

## Updating agent SSL certificates

For each agent that interacts with the PEM server, you must:

- Generate an rsa key and a certificate.
- Copy the key and certificate to the agent.
- Restart the agent.

Each agent has a unique identifier that's stored in the `pem.agent` table in the `pem` database. You must replace the key and certificate files with the key or certificate that corresponds to the agent's identifier. You must move the `agent.key` and `agent.crt` files generated in Steps 2 and 3 into place on their respective PEM agent host before generating the next key file pair. Subsequent commands overwrite the previously generated file.

To generate a PEM agent key file pair:

1. Use psql to find the number of agents and their corresponding identifiers:

```
/usr/pgsql-x/bin/psql -U postgres -d pem --no-psqlrc -t -A -c "SELECT ID FROM pem.agent"
```

- On Linux, you can also find the agent identifier and location of the keys and certificates in the `PEMAgent` section of the `/etc/postgres-reg.ini` file.
- On Windows, the information is stored in the registry under `HKEY_LOCAL_MACHINE\Software\EnterpriseDB\PEM\agent`.

2. After identifying the agents that need key files, generate an `agent.key` for each agent. To generate the key, execute the following command, capturing the output in a file:

```
/usr/pgsql-x/bin/psql -U postgres -d pem --no-psqlrc -t -A -c "SELECT openssl_rsa_generate_key(1024)"
> agent.key
```

Modify the privileges of the `agent.key` file:

```
chmod 600 agent.key
```

3. Generate a certificate for each agent. To generate a certificate, execute the following command, capturing the output in a certificate file:

```
/usr/pgsql-x/bin/psql -U postgres -d pem --no-psqlrc -t -A -c
"SELECT openssl_csr_to_cert(openssl_rsa_key_to_csr('${cat agent.key}',
'agent<$ID>', 'US', 'MA', 'Bedford', 'Postgres Enterprise Manager',
'support@enterprisedb.com'),
'/var/lib/pgsql/x/data/ca_certificate.crt',
'/var/lib/pgsql/x/data/ca_key.key')" > agent.crt
```

Where `$ID` is the agent number of the agent (retrieved by the `psql` command line).

4. Modify the privileges of the `agent.crt` file:

```
chmod 600 agent.crt
```

5. Replace each agent's key and certificate file with the newly generated files before restarting the PEM agent service:

On Linux with `init.d`, restart the service with the command:

```
/etc/init.d/pemagent start
```

On Linux with `systemd`, restart the service with the command:

```
systemctl start pemagent
```

On a Windows host, you can use the Services applet to start the PEM agent. The PEM agent service is named Postgres Enterprise Manager Agent. Select the service name in the Services dialog box and select **Start the service**.

## 19 Managing configuration settings

Postgres Enterprise Manager reads multiple configuration files at startup:

- `config.py` — This is the main configuration file. Don't modify it. You can use it as a reference for configuration settings that one of the following files might be override.
- `config_distro.py` — This file is read after `config.py` and is intended for packagers to change any settings that are required for their Postgres Enterprise Manager distribution. This typically includes certain paths and file locations. This file is optional. Packagers can create it in the same directory as `config.py` if it's needed.
- `config_local.py` — This file is read after `config_distro.py` and is intended for end users to change any default or packaging specific settings that they want to adjust to meet local preferences or standards. This file is optional. Create it in the same directory as `config.py` if it's needed.

A copy of the default `config.py` file is included in the PEM online help for reference.

## 20 Troubleshooting agent issues

You can troubleshoot these issues that can occur with the PEM agent.

## Restoring a deleted PEM agent

If an agent was deleted from the `pem.agent` table then you can't restore it. You need to use the `pemworker` utility to reregister the agent.

If an agent was deleted from the PEM web client but still has an entry in the `pem.agent` table with value of `active = f`, then you can restore the agent:

1. Check the values of the `id` and `active` fields:

```
pem=# SELECT * FROM pem.agent;
```

2. Update the status for the agent to `true` in the `pem.agent` table:

```
pem=# UPDATE pem.agent SET active=true WHERE id=<x>;
```

Where `x` is the identifier that was displayed in the output of the query used in step 1.

3. Refresh the PEM web client.

The deleted agent is restored. However, the servers that were bound to that agent might appear to be down. To resolve this issue, modify the PEM agent properties of the server to add the bound agent again. After the successful modification, the servers appear as running properly.

## Using the command line to delete a PEM agent with down or unknown status

Using the PEM web interface to delete PEM agents with Down or Unknown status can be difficult if the number of such agents is large. In this situation, you can use the command line interface to delete Down or Unknown agents.

Use the following query to delete the agents that are Down for more than `N` number of hours:

```
...
UPDATE pem.agent SET active=false WHERE id IN
(SELECT a.id FROM pem.agent
a JOIN pem.agent_heartbeat b ON (b.agent_id=a.id)
WHERE a.id IN
(SELECT agent_id FROM pem.agent_heartbeat WHERE (EXTRACT (HOUR FROM now())-
EXTRACT (HOUR FROM last_heartbeat)) > <N> ));
...
```

Use the following query to delete the agents with an Unknown status:

```
...
UPDATE pem.agent SET active=false WHERE id IN
(SELECT id FROM pem.agent WHERE id NOT IN
(SELECT agent_id FROM pem.agent_heartbeat));
...
```

## 21 Accessing the web interface

After installing a PEM server and agent, you can configure PEM to start monitoring and managing PostgreSQL or EDB Postgres Advanced Server instances. The PEM server installer installs the PEM web interface. You can use the interface to review information about objects that reside on monitored servers or to review statistical information gathered by the PEM server.

After the server installation completes, you can open the PEM interface in your choice of browser by navigating to:

```
https://<ip_address_of_PEM_host>:8443/pem
```

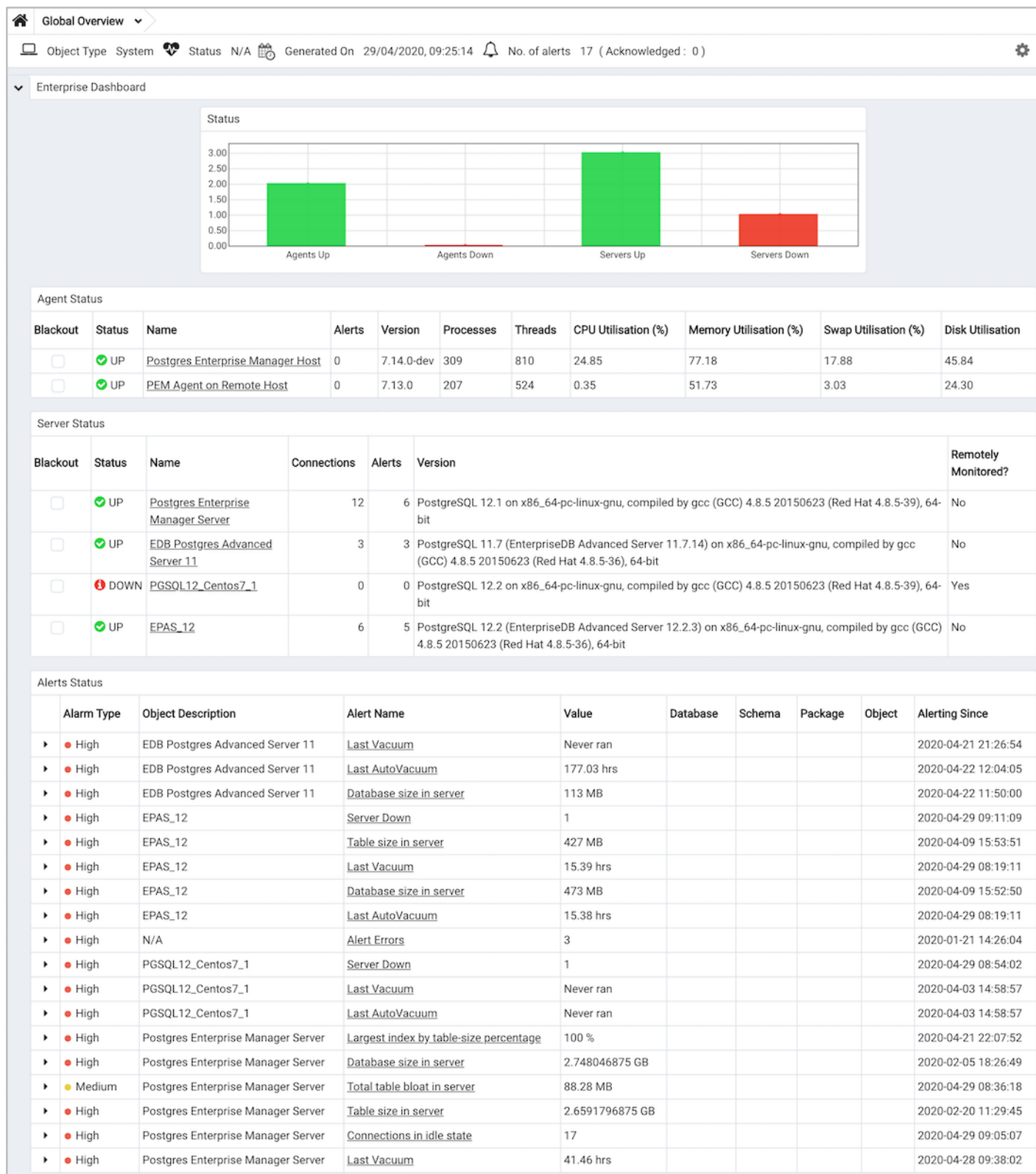
`ip_address_of_PEM_host` specifies the IP address of the host of the PEM server.

Use the fields on the Postgres Enterprise Manager Login window to authenticate yourself with the PEM server:

- Provide the name of a `pem` database user in the **Username** field. For the first user connecting, this is the name provided when installing the PEM server.
- Provide the password associated with the user in the **Password** field.

After providing your credentials, select **Login** to connect to PEM.

The PEM web interface opens, displaying the Global Overview dashboard.



The Browser pane displays a tree that provides access to information about the database objects that reside on each server. The tree expands to display a hierarchical view of the servers and objects that the PEM server monitors.

Before you can use the PEM web interface to manage or monitor a database server, you must register that server with the PEM server. When you register a server, you describe the connection to the server, provide authentication information for the connection, and specify any management preferences (optionally binding an agent).

A server can be managed or unmanaged:

- A *managed* server is bound to a PEM agent. The PEM agent monitors the server to which it's bound and performs tasks or reports statistics for display on the PEM dashboards. A managed server has access to extended PEM functionality such as custom alerting. When registering a server, you can



- also allow PEM to restart a managed server as required.
- An *unmanaged* server isn't bound to a PEM agent. You can create database objects on an unmanaged server, but extended PEM functionality (such as custom alerting) isn't supported on an unmanaged server.

You must also ensure the `pg_hba.conf` file of the server that you're registering allows connections from the host of the PEM web interface.

To access online help for the PEM web interface, select **Help** from the menu bar. The online help content is also available from the [EDB website](#).

The PEM menu bar

The PEM menu bar provides access to commands and features that you can use to manage your database servers and the objects that reside on those servers. If an option is disabled:

- The database server to which you are currently connected might not support the selected feature.
- The selected menu option might not apply to the current object.
- The role that you used to connect to the server might not have enough privileges to change the selected object.

Context-sensitive menus across the top of the PEM web interface allow you to customize your environment and provide access to the enterprise management features of PEM.

The File menu

Use the **File** menu to access the following options.

| Option               | Action  |
|----------------------|---|
| Preferences          | Open the Preferences dialog box to customize your PEM client settings.  |
| Lock Layout          | Open a submenu to select the level for locking the UI layout.   |
| Server Configuration | Open the Server Configuration dialog box and update your PEM server configuration settings.                                 |
| Reset Layout         | If a workspace panel is popped out by mistake or intentionally, you can reset it to the default using <b>Reset Layout</b> . |

The Object menu

The **Object** menu is context sensitive. Use the **Object** menu to access the following options.

| Option            | Action   |
|-------------------|--|
| Create            | Access a menu that provides context-sensitive selections.  |
| Refresh           | Refresh the currently selected object.   |
| Delete/Drop       | Delete the currently selected object from the server.  |
| Connect Server    | Open the Connect to Server dialog box to establish a connection with a server.   |
| CREATE Script     | Open the Query tool to edit or view the selected script.   |
| Disconnect Server | Refresh the currently selected object.   |
| BART              | Access a menu that provides options for removing BART configuration, taking a BART backup, or revalidating the BART configuration. |

| Option                    | Action   |
|---------------------------|--|
| Clear Saved Password      | If you saved the database server password, clear the saved password. Enabled only after password is saved.                         |
| Clear SSH Tunnel Password | If you saved the ssh tunnel password, clear the saved password. Enabled only after password is saved.                              |
| Drop Cascade              | Delete the currently selected object and all dependent objects from the server.  |
| Hide                      | Hide the currently selected group. To view hidden groups, enable the <b>Show hidden groups</b> option in Preferences.              |
| Properties                | Review or modify the currently selected object's properties.   |
| Trigger(s)                | Disable or enable triggers for the currently selected table.   |
| Truncate                  | Remove all rows from a table ( <b>Truncate</b> ) or remove all rows from a table and its child tables ( <b>Truncate Cascade</b> ). |
| View Data                 | Access a menu that provides several options for viewing data.  |

### The Management menu

Use the **Management** menu to access the following PEM features.

| Option                       | Action   |
|------------------------------|--|
| Audit Manager                | Open the Audit Manager and configure auditing on your monitored servers.                                       |
| Auto Discovery               | Open the Auto Discovery dialog box to configure a PEM agent to locate and bind monitored database servers.     |
| Capacity Manager             | Open the Capacity Manager dialog box and analyze historical or project future resource usage.                  |
| Log Manager                  | Open the Log Manager dialog box and configure log collection for a server.                                     |
| Manage Alerts                | Open the <b>Manage Alerts</b> tab and create or modify behavior for alerts.                                    |
| Manage Charts                | Open the <b>Manage Charts</b> tab to create or modify PEM charts.  |
| Manage Dashboards            | Open the Manage Dashboards dialog box to VACUUM, ANALYZE, REINDEX, or CLUSTER.                                 |
| Manage Probes                | Open the Manage Probes dialog box to VACUUM, ANALYZE, REINDEX, or CLUSTER.                                     |
| Postgres Expert              | Open the Postgres Expert wizard and perform a static analysis of your servers and databases.                   |
| Postgres Log Analysis Expert | Open the Postgres Log Analysis Expert dialog box to analyze log file contents for usage trends.                |
| Scheduled Tasks              | Open the <b>Scheduled Tasks</b> tab and review tasks that are pending or recently completed.                   |
| Tuning Wizard                | Open the Tuning Wizard dialog box to generate a set of tuning recommendations for your server.                 |
| Reports                      | Open the Reports dialog box to generate the system configuration report and core usage report for your server. |

### The Dashboards menu

Use the context-sensitive **Dashboards** menu to access dashboards.

| Option               | Action   |
|----------------------|--|
| Alerts               | Open the Alerts dashboard for the selected node.                   |
| Audit Log            | Open the Audit Log Analysis dashboard for the selected node.       |
| PGD Admin            | Open the PGD Admin dashboard for the selected node.                |
| PGD Group Monitoring | Open the PGD Group Monitoring dashboard for the selected node.     |
| PGD Node Monitoring  | Open the PGD Node Monitoring dashboard for the selected node.      |
| Database Server      | Open the Database Analysis dashboard for the selected node.        |
| I/O Analysis         | Open the I/O Analysis dashboard for the selected node.             |
| Memory               | Open the Memory Analysis dashboard for the selected node.          |
| Object Activity      | Open the Object Activity Analysis dashboard for the selected node. |

| Option                | Action   |
|-----------------------|--|
| Operating System      | Open the Operating System Analysis dashboard for the selected node.      |
| Probe Log             | Open the Probe Log Analysis dashboard for the selected node.             |
| Server Log            | Open the Server Log Analysis dashboard for the selected node.            |
| Session Activity      | Open the Session Activity Analysis dashboard for the selected node.      |
| Storage               | Open the Storage Analysis dashboard for the selected node.               |
| Streaming Replication | Open the Streaming Replication Analysis dashboard for the selected node. |
| System Wait           | Open the System Wait Analysis dashboard for the selected node.           |

### The Tools menu

Use the options on the **Tools** menu to access the following features.

| Option               | Action  |
|----------------------|---|
| Schema Diff          | Open the Schema Diff dialog box to compare the schema objects between two database schemas.   |
| Search objects       | Open the Search Objects dialog box to search the database objects in a database.  |
| Server               | Access the various server-related tools such as Add Named Restore Point, Performance Diagnostics, Queue Server Startup, Queue Server Shutdown, Replace Cluster Primary, Switchover EFM Cluster, and SQL Profiler. |
| Query tool           | Open the Query tool for the currently selected object.  |
| Reload Configuration | Update configuration files without restarting the server.   |
| Pause replay of WAL  | Pause replay of the WAL log.  |
| Resume replay of WAL | Resume replay of the WAL log.   |
| Import/Export        | Open the Import/Export Data dialog box to import or export data from a table.   |
| Maintenance          | Open the Maintenance dialog box to VACUUM, ANALYZE, REINDEX, or CLUSTER.  |
| Backup               | Open the Backup dialog box to back up database objects.   |
| Backup Globals       | Open the Backup Globals dialog box to back up cluster objects.  |
| Backup Server        | Open the Backup Server dialog box to back up a server.  |
| Restore              | Open the Restore dialog box to restore database files from a backup.  |
| Grant Wizard         | Open the Grant Wizard tool.   |
| Schedule Backup      | Open the Schedule Backup dialog box for BART backups.   |

### The Help menu







Use the options on the **Help** menu to access the online help documents or to review information about the PEM installation.

| Option                            | Action  |
|-----------------------------------|---|
| Online Help                       | Open documentation for Postgres Enterprise Manager.                     |
| REST API Reference                | Open the REST API Reference.  |
| EDB Website                       | Open the EDB website in a browser window.                               |
| About Postgres Enterprise Manager | Locate versioning and user information for Postgres Enterprise Manager. |

Controlling and customizing charts, graphs and tables

Use the icons in the upper-right corner of each graphic on a dashboard to control and customize the charts, graphs, and tables displayed in the PEM client for your current user session.

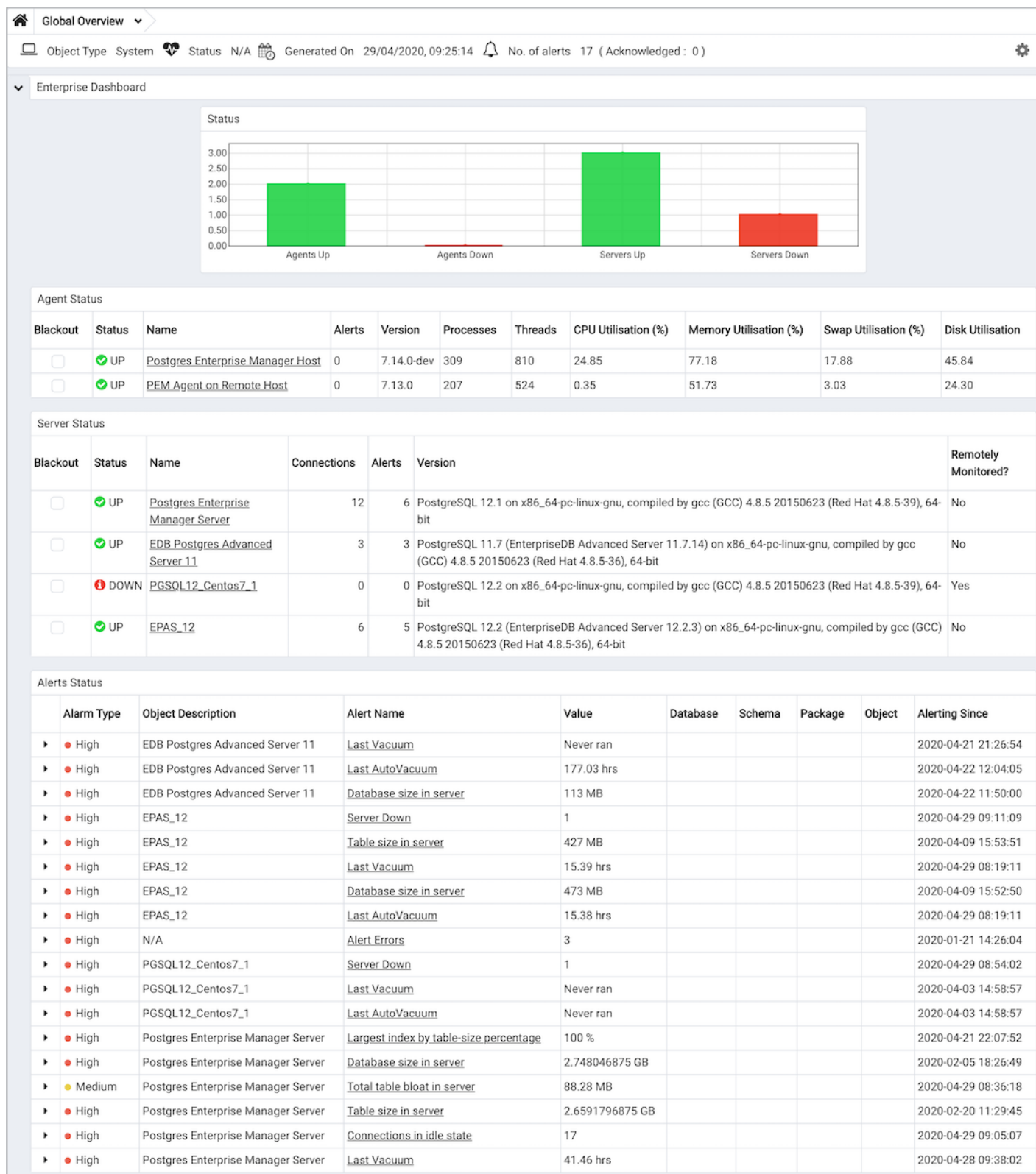
The following table gives the information about icons.

| Icons  | Information   |
|--|---|
|  | View information about the chart, graph, or table.                    |
|  | Refresh the content of a chart, graph, or table.                      |
|  | Personalize the chart, graph, or table settings for the current user. |
|  | Download an image of the chart or graph.                              |
|  | View the legends that are used in the chart, graph, or table.         |
|  | Expand the chart or graph to full-screen.                             |

For more information about customizing the graphics displayed on the PEM dashboards, see the PEM client online help.

22      Monitoring performance

PEM enables you to implement enterprise-wide performance monitoring of all managed servers.



The top-level dashboard is Global Overview. The Global Overview dashboard presents a status summary of all the servers and agents that are being monitored by the PEM server, a list of the monitored servers, and the state of any currently triggered alerts.

PEM provides a number of benefits not found in any other PostgreSQL management tool:

- Management en masse design — PEM is designed for enterprise database management and is built to tackle the management of large numbers of servers across geographical boundaries. Global dashboards visually keep you up to date on the up/down/performance status of all your servers.
- Distributed architecture — The PEM architecture maximizes the ability to gather statistical information and to perform operations remotely on machines regardless of operating system platform.

- Graphical administration — All aspects of database administration can be carried out in the PEM client using a graphical interface. You can handle server startup and shutdown, configuration management, storage and security control, object creation, performance management, and more from a single console.
- Full SQL IDE — PEM contains a robust SQL integrated development environment (IDE) that provides ad hoc SQL querying, stored procedure/function development, and a graphical debugger.
- Enterprise performance monitoring — PEM provides enterprise-class performance monitoring for all managed database servers. Lightweight and efficient agents monitor all aspects of each database server's operations as well as each machine's underlying operating system. They provide detailed statistics to easily navigate performance pages in the interface.
- Proactive alert management — PEM lets you create performance thresholds for each key metric, such as memory and storage, that are monitored around the clock. Any threshold violation results in an alert sent to a centralized dashboard that communicates the nature of the problem and the actions needed to prevent the situation from jeopardizing the overall performance of the server.
- Simplified capacity planning — All key performance-related statistics are collected and retained for a specified period in the PEM repository. The Capacity Manager utility allows you to select various statistics and perform trend analysis to understand things such as peak load periods and storage consumption trends. A forecasting mechanism in the tool allows you to also forecast resource usage so you can plan and budget accordingly.
- Audit Manager — Audit Manager configures audit logging on EDB Postgres Advanced Server instances. You can log activities such as connections to a database, disconnections from a database, and the SQL statements run against a database. You can then use the Audit Log dashboard to filter and view the log.
- Log Manager — The Log Manager wizard configures server logging parameters with (optional) log collection into a central table. Use the wizard to specify your preference for logging behaviors such as log file rotation, log destination, and the error message severity. Use the Server Log dashboard to filter and review the collected server log entries.
- SQL workload profiling — PEM contains a SQL profiling utility that allows you to trace the SQL statements that are executed against one or more servers. SQL profiling can be done either in an ad hoc or scheduled manner. You can then filter captured SQL statements so you can easily identify and tune SQL statements that are running poorly. You can also feed SQL statements into an index advisor on EDB Postgres Advanced Server that analyzes each statement and makes recommendations on new indexes to create to help performance.
- Expert database analysis — PEM includes the Postgres Expert utility. Postgres Expert analyzes selected databases for best practice enforcement purposes. Areas such as general configuration and security setup are examined. Any deviations from recommended best practices are reported back to you, along with an explanation of each issue and expert help on what to do to make things right.
- Streaming replication monitor — You can monitor the Streaming Replication dashboard or use options on the PEM client to promote a replica node to the primary node.
- Secure client connectivity — PEM supports secure client connections through an encrypted SSH tunnel. The full-featured PEM client includes an SSH Tunnel definition dialog box that allows you to provide connection information for a secure connection.
- Wide platform support — PEM supports most major Linux and Windows platforms.

## 22.1 Probes

A *probe* is a scheduled task that retrieves information about the database objects that are being monitored by the PEM agent. PEM uses the collected information to build the graphs displayed on each homepage. The **Management > Manage Probes** tab allows you to modify the data collection schedule and the length of time that PEM retains information returned by a specific probe.

### System probes

Unless otherwise noted, Postgres Enterprise Manager enables the following probes at the server, database, schema, extension (starting with version 8.4), or agent levels:

| Name                          | Information monitored by probe  | Level  |
|-------------------------------|---|--------|
| Background Writer Statistics  | Monitors information about the background writer. The information includes the number of:<br>Timed checkpoints<br>Requested checkpoints<br>Buffers written (by checkpoint)<br>Buffers written (by background writer)<br>Background writer cycles<br>Background buffers written<br>Buffers allocated   | Server |
| Barman Configuration          | Information about the Barman tool global configuration.   | Agent  |
| Barman Information            | Information about the Barman tool.  | Agent  |
| Barman Server                 | Information about the respective database server configuration monitored by Barman.   | Agent  |
| Barman Server Status          | Information about the respective database server status monitored by Barman.  | Agent  |
| Barman Server Backup          | Information about the backups of the respective database servers.   | Agent  |
| Barman Server WAL Status      | Information about the Barman server WAL files.  | Agent  |
| Blocked Session Information   | Information about blocked sessions.   | Server |
| CPU Usage                     | CPU usage information.  | Agent  |
| Data and Log File Analysis    | Information about log files. The information includes:<br>The name of the log file<br>The directory in which the log file resides   | Server |
| Database Frozen XID           | The frozen XID of each database.  | Server |
| Database Size                 | Information about the size of the monitored databases. The information includes:<br>The time the information was gathered<br>The database name<br>The database size (in MB).  | Server |
| Database Statistics           | Database statistics. The information includes:<br>The number of backends<br>The number of transactions committed<br>The number of transactions rolled back<br>The number of blocks read<br>The number of blocks hit<br>The number of rows returned<br>The number of rows fetched<br>The number of rows inserted<br>The number of rows updated<br>The number of rows deleted | Server |
| Disk Busy Info                | Information about disk activity.<br>Note: This probe isn't supported on Mac OS X, Solaris, or HP-UX.  | Agent  |
| Disk Space                    | Information about disk space usage. The information includes:<br>The amount of disk space used<br>The amount of disk space available  | Agent  |
| EDB Audit Configuration       | The audit logging configuration of EDB Postgres Advanced Server.  | Server |
| Failover Manager Cluster Info | Monitors a Failover Manager cluster, returning information about the cluster. This probe is enabled when a cluster name and path of the Failover Manager binary is provided on the Server Properties dialog box.  | Server |
| Failover Manager Node Status  | Monitors a Failover Manager cluster, returning details about each node within the cluster. This probe is enabled when a cluster name and path of the Failover Manager binary is provided on the Server Properties dialog box.   | Server |

| Name                            | Information monitored by probe  | Level    |
|---------------------------------|---|----------|
| Function Statistics             | Monitors a database, retrieving information about functions. The information includes:<br>Function names<br>Argument types<br>Return values   | Database |
| Index Size                      | Monitors a database, retrieving information about indexes. The information includes:<br>The name of the index<br>The time the data was gathered<br>The size of the index (in MB)  | Database |
| Index Statistics                | Index statistics. The information includes:<br>The number of index scans<br>The number of rows read<br>The number of rows fetched<br>The number of blocks read<br>The number of blocks hit  | Database |
| Installed Packages              | The packages that are currently installed. The information gathered includes:<br>The name of the installed package<br>The version of the installed package<br>The date and time that the probe executed   | Agent    |
| IO Analysis                     | Disk I/O information. The information includes:<br>The number of blocks read<br>The number of blocks written<br>The date and time that the probe executed<br>Note: This probe isn't supported on Mac OS X.  | Agent    |
| Load Average                    | CPU load averages. The information includes:<br>The 1-minute load average<br>The 5-minute load average<br>The 15-minute load average<br>Note: This probe isn't supported on Windows.  | Agent    |
| Lock Information                | Lock information. The information includes:<br>The database name<br>The lock type<br>The lock mode<br>The process holding the lock  | Server   |
| Memory Usage                    | Information about system memory usage. The information includes:<br>Total RAM in MB<br>Free RAM in MB<br>Total swap memory in MB<br>Free swap memory in MB<br>Shared system memory in MB (used by tuning wizard to tune the memory parameters for the database server)<br>- On non-Windows systems, it is the <code>shmmmax</code> value and read from <code>/proc/sys/kernel/shmmmax</code> .<br>- On Windows, it is same as total memory. | Agent    |
| Network Statistics              | Network statistics. The information includes:<br>The interface IP address<br>The number of packets sent<br>The number of packets received<br>The number of bytes sent<br>The number of bytes received<br>The link speed (in MB/second)  | Agent    |
| Number of Prepared Transactions | Stores the number of prepared transactions.   | Server   |
| Number of WAL Files             | The number of WAL files.  | Server   |
| Object Catalog: Database        | Monitors a list of databases and their properties. The information includes:<br>The database name<br>The database encoding type<br>If the database allows user connections or system connections  | Server   |



| Name                           | Information monitored by probe  | Level    |
|--------------------------------|---|----------|
| Object Catalog: Foreign Key    | Monitors a list of foreign keys and their properties. The information includes:<br>The name of the table that contains the foreign key<br>The name of the table that the foreign key references<br>The name of the database in which the table resides<br>The name of the schema in which the table resides | Schema   |
| Object Catalog: Function       | Monitors a list of functions and their properties. The information includes:<br>The name of the function<br>The name of the schema in which the function resides<br>The name of the database in which the function resides  | Schema   |
| Object Catalog: Index          | Monitors a list of indexes and their properties. The information includes:<br>The name of the index<br>The name of the table that the index is associated with<br>The name of the database in which the indexed table resides   | Schema   |
| Object Catalog: Schema         | Monitors a list of schemas and their associated databases and servers.  | Database |
| Object Catalog: Sequence       | Monitors a list of sequences and their properties.  | Schema   |
| Object Catalog: Table          | Monitors a list of table information. The information includes:<br>The table name<br>The name of the schema in which the table resides<br>The name of the database in which the schema resides<br>A Boolean indicator that shows whether the table has a primary key  | Schema   |
| Object Catalog: Tablespace     | Monitors a list of tablespaces.   | Server   |
| Operating System Information   | Operating system details and boot time.   | Agent    |
| Package Catalog                | The packages that are currently available for installation. The information gathered includes:<br>The package name<br>The package version   | Agent    |
| PG HBA Conf                    | Authentication configuration information from the <code>pg_hba.conf</code> file.  | Server   |
| Server Information             | Server information.   | Server   |
| Session Information            | Session information. The information includes:<br>The name of the session user<br>The date and time that the session connected to the server<br>The status of the session at the time that the information was gathered (idle, waiting, and so on)<br>The client address and port number                    | Server   |
| Settings                       | Monitors the values currently assigned to GUC variables.  | Server   |
| SQL Protect                    | Monitors a server, retrieving information about SQL injection attacks.  | Server   |
| Slony Replication              | Lag data for clusters replicated using Slony.   | Database |
| Streaming Replication          | Monitors a cluster that is using streaming replication, retrieving information about:<br>The sent Xlog location (in bytes)<br>The write Xlog location (in bytes)<br>The flush Xlog location (in bytes)<br>The replay Xlog location (in bytes)<br>The Xlog lag (in segments)<br>The Xlog lag (in pages)      | Server   |
| Streaming Replication Lag Time | Monitors a cluster that's using streaming replication, retrieving lag information about:<br>Replication lag time (in seconds)<br>Current status of replication (running/paused)   | Server   |

| Name                                     | Information monitored by probe  | Level    |
|--|---|----------|
| Streaming Replication Database Conflicts | Monitors a database that's using streaming replication, retrieving information about any conflicts that arise. This includes information about queries that were canceled due to the number of:<br>Drop tablespace conflicts<br>Lock timeout conflicts<br>Old snapshot conflicts<br>Pinned buffer conflicts<br>Deadlock conflicts   | Server   |
| Table Bloat                              | Information about the current table bloat. The information includes:<br>The name of the table<br>The name of the schema in which the table resides<br>The estimated number of pages<br>The estimated number of wasted pages<br>The estimated number of bytes per row  | Database |
| Table Frozen XID                         | Monitors the frozen XID of each table.  | Schema   |
| Table Size                               | Information about table size. The information includes:<br>Table size (in MB)<br>Total index size (in MB)<br>Total table size, with indexes and TOAST (in MB)   | Database |
| Table Statistics                         | Table statistics. The information includes:<br>The number of sequential scans<br>The number of sequential scan rows<br>The number of index scans<br>The number of index scan rows<br>The number of rows inserted<br>The number of rows updated<br>The number of rows deleted<br>The number of live rows<br>The number of dead rows<br>The last VACUUM<br>The last auto-vacuum<br>The last ANALYZE<br>The last auto-analyze<br>The number of pages estimated by ANALYZE<br>The number of rows estimated by ANALYZE | Database |
| Tablespace Size                          | A list of tablespaces and their sizes.  | Server   |
| User Information                         | A list of the current users. The stored information includes:<br>The user name<br>The user type (superuser or non-superuser)<br>The server to which the user is connected   | Server   |
| WAL Archive Status                       | Status of the WAL archive. The stored information includes:<br>The number of WAL archives done<br>The number of WAL archives pending<br>The last archive time<br>The number of WAL archives failed<br>The time of the last failure  | Server   |
| xDB Replication                          | Lag data for clusters replicated using xDB replication.   | Database |

## EDB Postgres Distributed (PGD) probes

To monitor the EDB Postgres Distributed group using dashboards, you must enable the following probes. Configure these probes at extension level.

## Note

Prior to version 8.4, all these probes are available at the server level.

The user with `pgd_superuser` can view information from all the following probes. If you are using a version of EDB Postgres Distributed earlier than 4.0, all these probes work with EDB Postgres Distributed Enterprise Edition.

| Probe name                         | Information monitored by probe   | <code>pgd_monitor</code> role required? | Works with PGD SE? |
|------------------------------------|--|---|--------------------|
| PGD Conflict History Summary       | Information about row conflicts per conflict type. The stored information includes:<br>The local time of the conflict<br>The type of the conflict  | Yes                                     | Yes                |
| PGD Global Locks                   | Information about global locks in a EDB Postgres Distributed group. The stored information includes:<br>The name of the node where the global lock originated<br>The PID of the process holding the lock<br>The type of lock (DDL or DML)<br>The name of the locked relation (for DML Locks) or keys (for advisory locks)<br>The internal state of the lock acquisition process<br>The list of backends waiting for the same global lock<br>The time when the global lock acquire was initiated by origin node<br>The time when the local node started trying to acquire the local lock<br>The time acquire_stage last changed   | Yes                                     | Yes                |
| PGD Group Camo Details             | Information about camos in a EDB Postgres Distributed group. The stored information includes:<br>The name of the node<br>The node name for whom this node is partner<br>The node name for whom this node is origin<br>The connection status<br>The readiness status<br>The number of pending or unresolved camo transactions<br>The LSN of last applied WAL log<br>The LSN of last received WAL log  | No                                      | No                 |
| PGD Group Replication Slot Details | Information about replication slots in a EDB Postgres Distributed group. The stored information includes:<br>The name of the EDB Postgres Distributed group<br>The name of the origin node<br>The name of the target node<br>The slot name on the origin node used by this subscription<br>The active status<br>The state of the replication (catchup,streaming, disconnected,...)<br>The approximate lag time for reported write<br>The approximate lag time for reported flush<br>The approximate lag time for reported replay<br>The bytes difference between sent_lsn and current WAL write position<br>The bytes difference between write_lsn and current WAL write position<br>The bytes difference between flush_lsn and current WAL write position | No                                      | Yes                |
| PGD Group Subscription Summary     | Information about the summary of subscriptions in the EDB Postgres Distributed group. The stored information includes:<br>The name of the origin of the subscription<br>The name of the target of the subscription<br>The timestamp of the last replayed transaction<br>The lag between now and time of last replayed transaction  | No                                      | Yes                |
| PGD Monitor Group Raft             | The status and message of a cluster-wide raft check.   | Yes                                     | Yes                |

| Probe name                 | Information monitored by probe  | pgd_monitor role required? | Works with PGD SE? |
|----------------------------|---|----------------------------|--------------------|
| PGD Group Raft Details     | Information about raft consensus status from all the nodes in a EDB Postgres Distributed group.<br>The stored information includes:<br>The name of the node<br>The raft worker state on the node<br>The node id of the RAFT_LEADER<br>The raft election internal id<br>The raft snapshot internal id  | Yes                        | Yes                |
| PGD Monitor Group Versions | The status and message of cluster-wide version check.   | Yes                        | Yes                |
| PGD Group Versions Details | Information about version details of the installed Postgres, pglogical, EDB Postgres Distributed for each node in the EDB Postgres Distributed group. The stored information includes:<br>The name of the node<br>The installed Postgres version on the node<br>The installed pglogical version on the node<br>The version of the EDB Postgres Distributed on the node<br>The EDB Postgres Distributed edition (Standard/Enterprise) for versions earlier than 4.0.   | No                         | Yes                |
| PGD Node Replication Rates | Information about outgoing replication activity from a given node. The stored information includes:<br>The name of the target peer node<br>The latest sent position<br>The latest position reported as replayed<br>The approximate lag time for reported replay<br>The bytes difference between replay_lsn and current WAL write position on origin<br>The human-readable bytes difference between replay_lsn and current WAL write position<br>Approximate time required for the peer node to catch up to all the changes that are yet to be applied   | Yes                        | No                 |
| PGD Node Slots             | Information about the mapping of local EDB Postgres Distributed database nodes to replication slots, their status, and replication progress. The stored information includes:<br>The name of the slot<br>The name of the target node<br>The name of the EDB Postgres Distributed group<br>The database name on the target node<br>The PID of the process attached to the slot<br>The catalog XID needed by the slot<br>The IP address of the client connection<br>The latest sent position<br>The latest position reported as replayed<br>The approximate lag time for reported replay<br>The bytes difference between replay_lsn and current WAL write position<br>The human-readable bytes difference between replay_lsn and current WAL write position | Yes                        | Yes                |
| PGD Node Summary           | Information about all the nodes in the EDB Postgres Distributed group. The stored information includes:<br>The name of the node<br>The name of the EDB Postgres Distributed group the node is part of<br>The consistent state of the node in human-readable form<br>The state which the node is trying to reach (during join or promotion)<br>The name of subscribed repsets  | Yes                        | Yes                |

| Probe name      | Information monitored by probe  | pgd_monitor role required? | Works with PGD SE? |
|-----------------|---|----------------------------|--------------------|
| PGD Workers     | Information about workers in a EDB Postgres Distributed node. The stored information includes:                | Yes                        | Yes                |
|                 | The PID of the worker process   |                            |                    |
|                 | The worker query start timestamp  |                            |                    |
|                 | The worker state change timestamp   |                            |                    |
|                 | The worker wait event type  |                            |                    |
|                 | The worker wait event   |                            |                    |
|                 | The worker state  |                            |                    |
|                 | The worker role name  |                            |                    |
|                 | The worker commit timestamp   |                            |                    |
|                 | The worker local timestamp  |                            |                    |
|                 | The name of the origin node   |                            |                    |
|                 | The receive LSN   |                            |                    |
|                 | The receive commit LSN  |                            |                    |
|                 | The last exact replay LSN   |                            |                    |
|                 | The last exact flush LSN  |                            |                    |
|                 | The last exact replay timestamp   |                            |                    |
|                 | The worker query  |                            |                    |
| PGD Work Errors | Information about the work errors in EDB Postgres Distributed database node. The stored information includes: | Yes                        | Yes                |
|                 | The process id of the worker causing the error  |                            |                    |
|                 | The name of the EDB Postgres Distributed group the node is part of  |                            |                    |
|                 | The name of the origin node   |                            |                    |
|                 | The name of the source node   |                            |                    |
|                 | The name of the target node   |                            |                    |
|                 | The name of the subscription  |                            |                    |
|                 | The internal identifier of the role of this worker  |                            |                    |
|                 | The name of the role of this worker   |                            |                    |
|                 | The date and time of the error  |                            |                    |
|                 | The age of the error  |                            |                    |
|                 | The description of the error  |                            |                    |
|                 | The context in which the error happened   |                            |                    |
|                 | The remote relation id  |                            |                    |
|                 | The subscription writer id  |                            |                    |
|                 | The subscription writer name  |                            |                    |

Customizing probes

A probe is a scheduled task that returns a set of performance metrics about a specific monitored object. A probe retrieves statistics from a monitored server, database, operating system, or agent. You can use the **Manage Probes** tab to override the default configuration and customize the behavior of each probe.

To open the **Manage Probes** tab, select **Management > Manage Probes**. The **Manage Probes** tab provides a set of icons that you can use to create and manage probes:

- Select **Manage Custom Probes** to open the **Custom Probes** tab and create or modify a custom probe.
- Select **Copy Probes** to open the Copy Probe dialog box and copy the probe configurations from the currently selected object to one or more monitored objects.

Note

Copy Probe isn't supported for the extension-level probes.

A probe monitors a unique set of metrics for each object type (server, database, database object, or agent). Select the name of an object in the tree to review the probes for that object.

To modify the properties associated with a probe, select the name of a probe and customize the settings that are displayed in the Probes table:

- Move the **Default** switch in the Execution Frequency columns to **N** to enable the **Minutes** and **Seconds** selectors, and specify a custom value for the length of time between executions of the probe.
- Move the **Default** switch in the Enabled? column to **No** to change the state of the probe and indicate if the probe is active or not.

### Note

If data from a disabled probe is used in a chart, the chart displays an information icon in the upper-left corner that allows you to enable the probe by clicking the provided link.

- Move the **Default** switch in the Data Retention column to **No** to enable the **Day(s)** field and specify the number of days that information gathered by the probe is stored on the PEM server.

The **Manage Probes** tab might display information about probes that you can't modify from the current node. If you can't modify a probe from the current dialog box, the switches are disabled. Generally, you can modify a disabled probe from a node that's higher in the hierarchy of the PEM client tree control. Select another object in the tree control to modify the probes that are displayed or enabled in the **Manage Probes** tab.

### Creating a custom probe

You can use the **PEM Custom Probes** tab to create a new probe or modify an existing custom probe. To open the **Custom Probes** tab, from the **Manage Probes** tab, select **Manage Custom Probes**.

Use the **Show System Probes?** switch to show or hide the system probes on the **Custom Probes** tab.

You can use the **Custom Probes** tab to create a probe or modify an existing one. To create a probe:

1. Select **Add** in the upper-right corner of the tab.
2. Provide a name for the new probe in the Probe Name column.
3. Select **Edit** (located to the left of the probe name) to review or add the probe definition.

Use the **General** tab to modify the definition of an existing probe or to specify the properties of a new probe:

- Use the **Probe Name** field to provide a name for a new probe.
- Use the **Collection method** field to specify the probe type. From the list, select:
  - SQL — The probe gathers information by way of a SQL statement.
  - WMI — The probe gathers information by way of a Windows Management Instrumentation extension.
  - Batch — The probe uses a command-script or shell-script to gather information.

Before creating a batch probe on a Linux system, you must modify the `agent.cfg` file, setting the `allow_batch_probes` parameter to `true`. Then restart the PEM agent. The `agent.cfg` file is located in one of the following directories:

- If you installed PEM using graphical installer: `/opt/edb/pem/agent/etc/agent.cfg`
- If you installed PEM using RPM: `/usr/edb/pem/agent/etc/agent.cfg`

On Windows systems, agent settings are stored in the registry. Before creating a batch probe, modify the registry entry for the `AllowBatchProbes` registry entry and restart the PEM agent. PEM registry entries are located in `HKEY_LOCAL_MACHINE\Software\EnterpriseDB\PEM\agent`.

Batch probes are platform specific. If you specify a collection method of **Batch**, you must specify a platform type in the **Platform** field.

To invoke a script on a Linux system, you must modify the entry for the `batch_script_user` parameter of the `agent.cfg` file and specify the user to use to run the script. You can either specify a non-root user or root for this parameter. If you don't specify a user or the specified user doesn't exist, then the script doesn't run. Restart the agent after modifying the file.

To invoke a script on a Windows system, set the registry entry for `AllowBatchJobSteps` to `true` and restart the PEM agent.

- Use the **Target Type** list to select the object type for the probe to monitor. **Target Type** is disabled if **Collection method** is **WMI**.
- Use the **Minutes** and **Seconds** selectors to specify how often the probe collects data.
- Use the **Probe Enable?** switch to specify if the probe is enabled by default. Specify **Yes** to enable the probe by default or **No** to disable the probe by default.

#### Note

If data from a disabled probe is used in a chart, the chart displays an information icon in the upper-left corner that allows you to enable the probe by clicking the provided link.

- Use the **Data Retention** field to specify the number of days that gathered information is retained in the probe's history table.
- Use the switch next to **Discard from history** to specify if the server creates a history table for the probe. Select **Yes** to discard probe history or **No** to retain the probe history in a table.
- Use the **Platform** list to specify the type of platform that the probe monitors. This field is enabled only when the **Collection method** is **Batch**.

Use the **Columns** tab to define the columns in which to store the probe data. To define a new column:

1. Navigate to the **Columns** tab and select **Add** in the upper-right corner.
2. Provide a column name in the **Name** field.
3. Select **Edit** (to the left of the new column name) to provide information about the column:

- Provide a descriptive name for the column in the **Name** field.
- The **Internal Name** field isn't enabled for custom probes.
- Use the **Column Type** list to specify if the column is a key column (a primary key) or a non-key column. Non-key columns are generally metric items (values that can be graphed).
- Use the **Data Type** list to specify the type of data to store in the column.
- Use the **Unit** field to specify the unit of measure that applies to the metric stored in the column. This unit is displayed on the y-axis of a custom chart or a Capacity Manager chart. This is an optional field.
- Use the **Graphable** switch to specify if the defined metric can be graphed and to make the probe accessible from the Capacity Manager or Manage Charts dialog boxes.
- Use the **Is PIT** switch to specify whether to store the metric by point-in-time.

*Point-in-time* metrics change (increase or decrease) at any given point of time. For example, database size is a point-in-time metric. At any given point-in-time, the size of the database is fluctuating. Metrics that aren't point-in-time (also referred to as cumulative metrics) are those whose size always increases over time. For example, Blocks Read and Tuples Read are cumulative metrics. The value stays the same or increases.

- Use the **Calculate PIT** switch to specify for the server to calculate a point-in-time value for the metric data. **Calculate PIT** is disabled if **Is PIT** is **Yes**.

PEM allows you to store point-in time-values of cumulative metrics as well. PEM subtracts the last collected value of a cumulative metric from the current value and stores the difference as a point-in-time value.

Use the **Code** tab to specify the default code for the probe to execute:

- If the probe is a SQL probe, you must specify the **SQL SELECT** statement invoked by the probe on the **Code** tab. The column names returned by the query must match the **Internal Name** specified on the **Columns** tab. The number of columns returned by the query, as well as the column name, data type, and so on, must match the information specified on the **Columns** tab.
- If the probe is a batch probe, you must specify the shell or .bat script to invoke when the probe runs. The output of the script is as follows:
  - The first line must contain the names of the columns provided on the **Columns** tab. Each column name must be separated by a tab (t) character. From the second line onwards, each line must contain the data for each column, separated by a tab character.

- If a specified column is defined as a key column, make sure that the script doesn't produce duplicate data for that column across lines of output. The number of columns specified in the **Columns** tab and their names, data type, and so on must match with the output of the script output.
- If the probe is a WMI probe, you must specify the WMI query as a **SELECT WMI** query. The column name referenced in the **SELECT** statement must be same as the name of the corresponding column specified on the **Column** tab. The column names returned by the query must match the **Internal Name** specified on the **Columns** tab. The number of columns returned by the query, as well as the column name, data type, and so on, must match the information specified on the **Columns** tab.

Use the **Alternate Code** tab to provide code to invoke if the probe fires on a specific version of the server. To provide version-specific code, move the **Applies to any server version?** switch to **No** and select **Add**. Then, use the **Database Version(s)** list to select a version, and select **Edit** (to the left of the version name) to provide the code to execute when the probe fires.

If you select a database version and leave the Probe Code column blank, PEM invokes the code specified on the **Code** tab when the probe executes on a server that matches that version.

When you've finished defining the probe, select **Save** (in the corner of the **Custom Probes** tab) to save the definition and make the probe data available for use on custom charts and graphs.

## Exporting or importing a probe

### New Feature

Export and import of probes to another PEM Server is available from 8.2.0 and later.

To export the probe, select any probes from the **Manage Custom Probes** tab, and then select **Export** in the upper-right corner of the table. To generate the JSON file, select **Save File** and then select **OK**.

To import the probe, on the **Manage Custom Probes** tab, select **Import** in the upper-right corner of the table. Select **Browse** to select the JSON file with the probe code to import, and then select **Import**.

After selecting the file to import, you can select the **Skip existing** check box. This option skips the probe if it already exists and displays a message letting you know that the import was skipped.

- If you don't select the check box and the probe already exists, then it doesn't import the probe and an error message is reported.

!!! Note **Import** can't overwrite the existing probe as it might be configured to retain historical data as per the configured retention policy.

- If you don't select the check box and the probe doesn't exist but the corresponding table in the **pem** schema does, then it imports the probe successfully using the same table.

### Note

It's possible for the probe to be deleted and not listed on **Manage Custom Probe** tab and for the table holding the data of that probe to exist in the **pem** schema.

## Deleting a probe

You can delete only custom probes. To delete a probe, select the probe name in the probes table, and select **Delete** (located to the upper-right corner of the table). The probe history persists for the length of time specified in the **History Retention** field in the probe definition. During the deletion, the probe definition is deleted and any corresponding tables are dropped from the **pemdata** and **pemhistory** schemas.

System probes are the built-in probes provided by PEM and are part of the PEM schema. You can only modify system probes. If you attempt to delete a system probe, PEM reports an error.



## Copying a probe

You can use the Copy Probe Configuration dialog box to copy probe definitions from one monitored object to one or more monitored objects of the same type. To open the Copy Probe Configuration dialog box:

1. On the PEM client tree, select the object from which you are copying probes.
2. Select **Management > Manage Probes**.
3. From the **Manage Probes** tab, select **Copy Probe**.

The dialog box copies the probe definitions from the object from which you opened the Copy Probe Configuration dialog box to the locations selected on the tree control.

If you specify a parent node in the **Copy Probe Configuration** tree, PEM copies the probe configurations to each object of the same type that resides under that node in the tree. For example, to copy the probe definitions from one schema to all schemas that reside in a database, select only the parent database of the target schemas. A red warning symbol is displayed to the left of the name of a listed target object if that object is the source of the probe that's being copied.

After you select the target object or objects, select **Configure Probes** to copy the probe definitions to the location selected in the dialog box.

## PEM probes – reference

A probe is a scheduled task that retrieves information about the database objects that are being monitored by the PEM agent. PEM uses the collected information to build the graphs displayed on each dashboard. The **Manage Probes** tab (accessed from the **Management** menu) allows you to modify the data collection schedule and the length of time that PEM retains information returned by a specific probe.

| Name                         | Information monitored  | Level  |
|------------------------------|--|--------|
| Background Writer Statistics | Information about the background writer. The information includes:<br>The number of timed checkpoints<br>The number of requested checkpoints<br>The number of buffers written (by checkpoint)<br>The number of buffers written (by background writer)<br>The number of background writer cycles<br>The number of background buffers written<br>The number of buffers allocated | Server |
| Blocked Session Information  | Information about blocked sessions.  | Server |
| CPU Usage                    | CPU Usage information.   | Agent  |
| Data and Log File Analysis   | Information about log files. The information includes:<br>The name of the log file<br>The directory in which the log file resides  | Server |
| Database Frozen XID          | The frozen XID of each database.   | Server |
| Database Size                | Information about the size of the monitored databases. The information includes:<br>The time the information was gathered<br>The database name<br>The database size (in MB).   | Server |

| Name                          | Information monitored   | Level    |
|-------------------------------|---|----------|
| Database Statistics           | Database statistics. The information includes the number of:<br>Backends<br>Transactions committed<br>Transactions rolled back<br>Blocks read<br>Blocks hit<br>Rows returned<br>Rows fetched<br>Rows inserted<br>Rows updated<br>Rows deleted | Server   |
| Disk Busy Info                | Information about disk activity.<br>Note: This probe isn't supported on Mac OS X, Solaris, or HP-UX.  | Agent    |
| Disk Space                    | Information about disk space usage. The information includes:<br>The amount of disk space used<br>The amount of disk space available  | Agent    |
| EDB Audit Configuration       | The audit logging configuration of EDB Postgres Advanced Server.  | Server   |
| Failover Manager Cluster Info | Monitors a Failover Manager cluster, returning information about the cluster. This probe is enabled when you provide a cluster name and path of the Failover Manager binary in the Server Properties dialog box.                              | Server   |
| Failover Manager Node Status  | Monitors a Failover Manager cluster, returning detailed about each node in the cluster. This probe is enabled when you provide a cluster name and path of the Failover Manager binary in the Server Properties dialog box.                    | Server   |
| Function Statistics           | Monitors a database, retrieving information about functions. The information includes:<br>Function names<br>Argument types<br>Return values   | Database |
| Index Size                    | Monitors a database, retrieving information about indexes. The information includes:<br>The name of the index<br>The time the data was gathered<br>The size of the index (in MB)  | Database |
| Index Statistics              | Index statistics. The information includes the number of:<br>Index scans<br>Rows read<br>Rows fetched<br>Blocks read<br>Blocks hit  | Database |
| Installed Packages            | The packages that are currently installed. The information gathered includes:<br>The name of the installed package<br>The version of the installed package<br>The date and time that the probe executed                                       | Agent    |
| IO Analysis                   | Monitors disk I/O information. The information includes:<br>The number of blocks read<br>The number of blocks written<br>The date and time that the probe executed<br>Note: This probe isn't supported on Mac OS X.                           | Agent    |
| Load Average                  | CPU load averages. The information includes:<br>The 1-minute load average<br>The 5-minute load average<br>The 15-minute load average<br>Note: This probe isn't supported on Windows.  | Agent    |
| Lock Information              | Monitors lock information. The information includes:<br>The database name<br>The lock type<br>The lock mode<br>The process holding the lock   | Server   |

| Name                            | Information monitored  | Level    |
|---------------------------------|--|----------|
| Memory Usage                    | Information about system memory usage. The information includes:<br>Total RAM in MB<br>Free RAM in MB<br>Total swap memory in MB<br>Free swap memory in MB<br>Shared system memory in MB (used by tuning wizard to tune the memory parameters for the database server)<br>- On non-Windows system, it's the <code>shmmmax</code> value and read from <code>/proc/sys/kernel/shmmmax</code> .<br>- On Windows, it's same as total memory. | Agent    |
| Network Statistics              | Network statistics. The information includes:<br>The interface IP address<br>The number of packets sent<br>The number of packets received<br>The number of bytes sent<br>The number of bytes received<br>The link speed (in MB/second)   | Agent    |
| Number of Prepared Transactions | Stores the number of prepared transactions.  | Server   |
| Number of WAL Files             | The number of WAL files.   | Server   |
| Object Catalog: Database        | Monitors a list of databases and their properties. The information includes:<br>The database name<br>The database encoding type<br>If the database allows user connections or system connections   | Server   |
| Object Catalog: Foreign Key     | Monitors a list of foreign keys and their properties. The information includes:<br>The name of the table that contains the foreign key<br>The name of the table that the foreign key references<br>The name of the database in which the table resides<br>The name of the schema in which the table resides  | Schema   |
| Object Catalog: Function        | Monitors a list of functions and their properties. The information includes:<br>The name of the function<br>The name of the schema in which the function resides<br>The name of the database in which the function resides   | Schema   |
| Object Catalog: Index           | Monitors a list of indexes and their properties. The information includes:<br>The name of the index<br>The name of the table that the index is associated with<br>The name of the database in which the indexed table resides  | Schema   |
| Object Catalog: Schema          | Monitors a list of schemas and their associated databases and servers.   | Database |
| Object Catalog: Sequence        | Monitors a list of sequences and their properties.   | Schema   |
| Object Catalog: Table           | Monitors a list of table information. The information includes:<br>The table name<br>The name of the schema in which the table resides<br>The name of the database in which the schema resides<br>A Boolean indicator that shows whether the table has a primary key   | Schema   |
| Object Catalog: Tablespace      | Monitors a list of tablespaces.  | Server   |
| Operating System Information    | The operating system details and boot time.  | Agent    |
| Package Catalog                 | The packages that are currently available for installation. The information gathered includes:<br>The package name<br>The package version  | Agent    |
| PG HBA Conf                     | Authentication configuration information from the <code>pg_hba.conf</code> file.   | Server   |

| Name                                     | Information monitored   | Level    |
|--|---|----------|
| Server Information                       | Server information.   | Server   |
| Session Information                      | Session information. The information includes:<br>The name of the session user<br>The date and time that the session connected to the server<br>The status of the session at the time that the information was gathered (idle, waiting, and so on)<br>The client address and port number  | Server   |
| Settings                                 | The values currently assigned to GUC variables.   | Server   |
| SQL Protect                              | Monitors a server, retrieving information about SQL injection attacks.  | Server   |
| Slony Replication                        | Lag data for clusters replicated using Slony.   | Database |
| Streaming Replication                    | Monitors a cluster that's using streaming replication, retrieving information about:<br>The sent Xlog location (in bytes)<br>The write Xlog location (in bytes)<br>The flush Xlog location (in bytes)<br>The replay Xlog location (in bytes)<br>The Xlog lag (in segments)<br>The Xlog lag (in pages)                             | Server   |
| Streaming Replication Lag Time           | Monitors a cluster that's using streaming replication, retrieving lag information about:<br>Replication lag time (in seconds)<br>Current status of replication (running/paused)   | Server   |
| Streaming Replication Database Conflicts | Monitors a database that's using streaming replication, retrieving information about any conflicts that arise. This includes information about queries that were canceled due to the number of:<br>Drop tablespace conflicts<br>Lock timeout conflicts<br>Old snapshot conflicts<br>Pinned buffer conflicts<br>Deadlock conflicts | Server   |
| Table Bloat                              | Information about the current table bloat. The information includes:<br>The name of the table<br>The name of the schema in which the table resides<br>The estimated number of pages<br>The estimated number of wasted pages<br>The estimated number of bytes per row  | Database |
| Table Frozen XID                         | The frozen XID of each table.   | Schema   |
| Table Size                               | Information about table size. The information includes:<br>Table size (in MB)<br>Total index size (in MB)<br>Total table size, with indexes and TOAST (in MB)   | Database |

| Name               | Information monitored   | Level    |
|--------------------|---|----------|
| Table Statistics   | Table statistics. The information includes:<br>The number of sequential scans<br>The number of sequential scan rows<br>The number of index scans<br>The number of index scan rows<br>The number of rows inserted<br>The number of rows updated<br>The number of rows deleted<br>The number of live rows<br>The number of dead rows<br>The last VACUUM<br>The last auto-vacuum<br>The last ANALYZE<br>The last auto-analyze<br>The number of pages estimated by ANALYZE<br>The number of rows estimated by ANALYZE | Database |
| Tablespace Size    | A list of tablespaces and their sizes.  | Server   |
| User Information   | A list of the current users. The stored information includes:<br>The user name<br>The user type (superuser or non-superuser)<br>The server to which the user is connected   | Server   |
| WAL Archive Status | The status of the WAL archive. The stored information includes:<br>The number of WAL archives done<br>The number of WAL archives pending<br>The last archive time<br>The number of WAL archives failed<br>The time of the last failure  | Server   |
| xDB Replication    | Monitors lag data for clusters replicated using xDB replication.  | Database |

22.2 Alerts

PEM continually monitors registered servers. It compares performance metrics against predefined and user-specified thresholds that specify good or acceptable performance for each statistic. Any deviation from an acceptable threshold value triggers an alert. An alert is a system-defined or user-defined set of conditions that PEM compares to the system statistics. Alerts tell you about conditions on registered servers that require your attention.

Viewing the alerts via Global dashboard

When your system statistics deviate from the boundaries specified for that statistic, the alert triggers. The alert displays a high (red), low (yellow), or medium (orange) severity warning in the left-most column of the Alert Status table on the Global Overview dashboard.

| Alerts Status |            |                                    |  |                 |          |        |         |        |                     |
|---------------|------------|------------------------------------|--|-----------------|----------|--------|---------|--------|---------------------|
|               | Alarm Type | Object Description                 | Alert Name   | Value           | Database | Schema | Package | Object | Alerting Since      |
| ▶             | ● High     | EDB Postgres Advanced Server 11    | <a href="#">Database size in server</a>                | 113 MB          |          |        |         |        | 2020-04-22 11:50:00 |
| ▶             | ● High     | EDB Postgres Advanced Server 11    | <a href="#">Last Vacuum</a>                            | Never ran       |          |        |         |        | 2020-04-21 21:26:54 |
| ▶             | ● High     | EDB Postgres Advanced Server 11    | <a href="#">Last AutoVacuum</a>                        | 140.21 hrs      |          |        |         |        | 2020-04-22 12:04:05 |
| ▶             | ● High     | EPAS_12                            | <a href="#">Table size in server</a>                   | 410 MB          |          |        |         |        | 2020-04-09 15:53:51 |
| ▶             | ● Medium   | EPAS_12                            | <a href="#">Last Vacuum</a>                            | 5.18 hrs        |          |        |         |        | 2020-04-27 20:47:50 |
| ▶             | ● High     | EPAS_12                            | <a href="#">Database size in server</a>                | 455 MB          |          |        |         |        | 2020-04-09 15:52:50 |
| ▶             | ● Medium   | EPAS_12                            | <a href="#">Last AutoVacuum</a>                        | 5.16 hrs        |          |        |         |        | 2020-04-27 20:47:50 |
| ▶             | ● High     | N/A                                | <a href="#">Alert Errors</a>                           | 3               |          |        |         |        | 2020-01-21 14:26:04 |
| ▶             | ● High     | PGSQL12_Centos7_1                  | <a href="#">Server Down</a>                            | 1               |          |        |         |        | 2020-04-27 20:48:50 |
| ▶             | ● High     | PGSQL12_Centos7_1                  | <a href="#">Last Vacuum</a>                            | Never ran       |          |        |         |        | 2020-04-03 14:58:57 |
| ▶             | ● High     | PGSQL12_Centos7_1                  | <a href="#">Last AutoVacuum</a>                        | Never ran       |          |        |         |        | 2020-04-03 14:58:57 |
| ▶             | ● High     | Postgres Enterprise Manager Server | <a href="#">Largest index by table-size percentage</a> | 100 %           |          |        |         |        | 2020-04-21 22:07:52 |
| ▶             | ● High     | Postgres Enterprise Manager Server | <a href="#">Database size in server</a>                | 2.072265625 GB  |          |        |         |        | 2020-02-05 18:26:49 |
| ▶             | ● High     | Postgres Enterprise Manager Server | <a href="#">Table size in server</a>                   | 1.9814453125 GB |          |        |         |        | 2020-02-20 11:29:45 |
| ▶             | ● Medium   | Postgres Enterprise Manager Server | <a href="#">Connections in idle state</a>              | 12              |          |        |         |        | 2020-04-27 16:20:32 |
| ▶             | ● Medium   | Postgres Enterprise Manager Server | <a href="#">Last Vacuum</a>                            | 4.99 hrs        |          |        |         |        | 2020-04-27 20:47:50 |

The PEM server includes a number of predefined alerts that are actively monitoring your servers. The alert definition might make details available about the cause of the alert. Select the down arrow to the right of the severity warning to open a dialog box that has details about the condition that triggered the alert.

Alert Details (Auto-refresh paused whilst rows are expanded. ⓘ)

|   | Ack'd                    | Alert Type | Name                                 | Value           | Agent | Server                             | Database | Schema | Package | Object | Alerting Since      |
|---|--------------------------|------------|--------------------------------------|-----------------|-------|------------------------------------|----------|--------|---------|--------|---------------------|
| ▶ | <input type="checkbox"/> | ● High     | <a href="#">Table size in server</a> | 1.9814453125 GB |       | Postgres Enterprise Manager Server |          |        |         |        | 2020-02-20 11:29:45 |

General

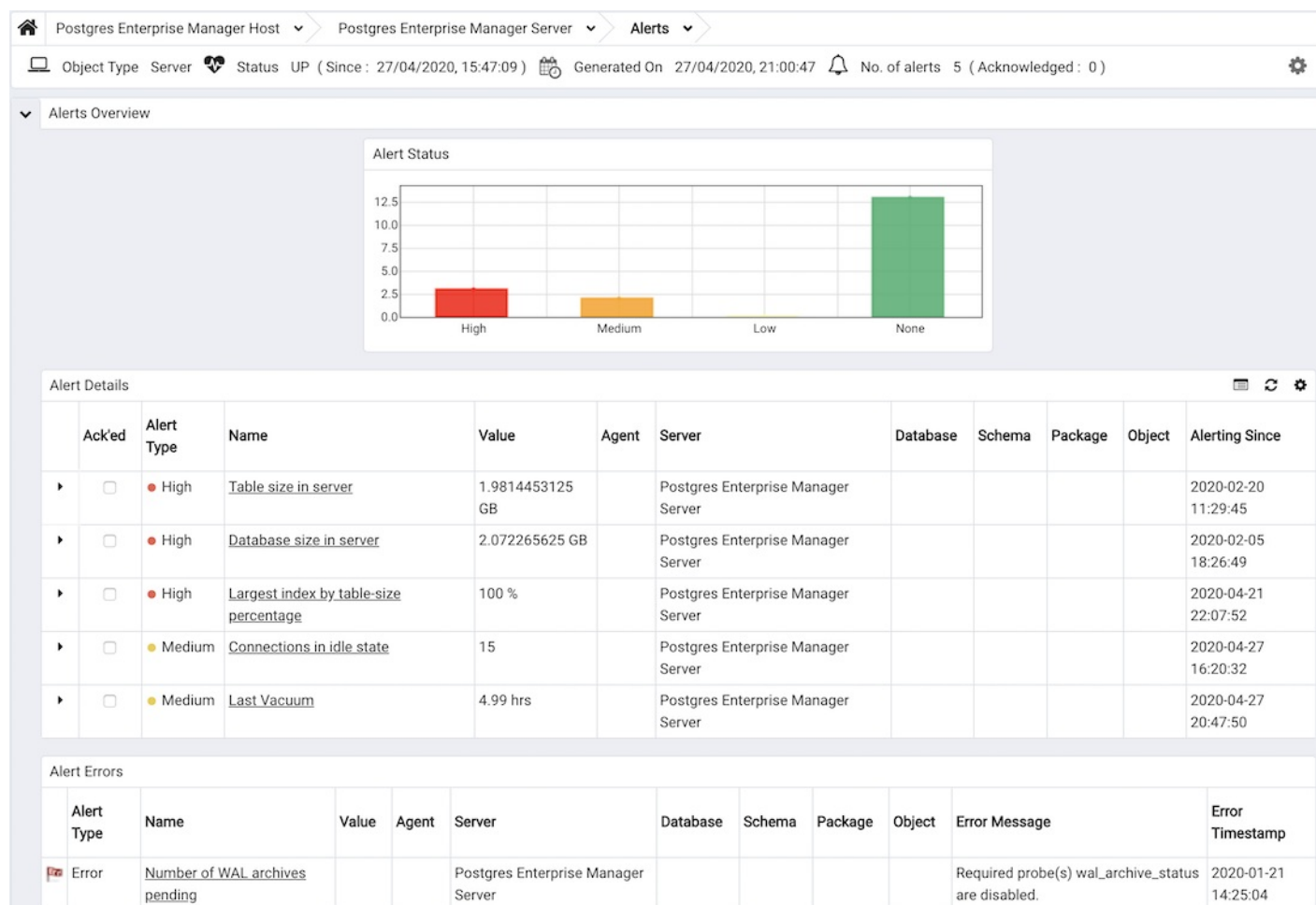
Parameters

| Table name       | Schema name | Database name | Total table size(MB) |
|------------------|-------------|---------------|----------------------|
| table_statistics | pemhistory  | pem           | 1087                 |
| server_logs      | pemdata     | pem           | 263                  |
| index_statistics | pemhistory  | pem           | 237                  |
| session_info     | pemhistory  | pem           | 137                  |
| lock_info        | pemhistory  | pem           | 88                   |

PEM also provides an interface that lets you create customized alerts. Each alert uses metrics defined on an alert template. An alert template defines how the server evaluates the statistics for a resource or metric. The PEM server includes predefined alert templates, and you can create custom alert templates.

### Viewing the alerts via Alerts dashboard

Use the **Dashboards** menu (on the **Monitoring** tab) to open the Alerts dashboard. The Alerts dashboard shows a summary of the active alerts and the status of each alert.



The Alerts dashboard header shows the date and time that the dashboard was last updated and the number of current alerts.

The Alerts Overview section shows a visual representation of the active alerts and a count of the current high, low, and medium alerts. The vertical bar on the left of the graph provides the count of the alerts displayed in each column. Hover over a bar to display the alert count for the selected alert severity in the upper-right corner of the graph.

The Alert Details table provides a list of the alerts that are currently triggered. The entries appear in order from high severity to low severity. Each entry includes information that lets you identify the alert and recognize the condition that triggered the alert. Select an alert to review detailed information about the alert definition.

The Alert Errors table shows configuration-related errors, such as accidentally disabling a required probe or improperly configuring an alert parameter. You can use the information provided in the Error Message column to identify and resolve the conflict that's causing the error.

### Customizing the Alerts dashboard

You can customize tables and charts that appear on the Alerts dashboard. To customize a table or chart, select **Settings** in the upper-right corner.

Use fields on the Personalize Chart Configuration dialog box to provide your display preferences:

- Use the **Auto Refresh** field to specify the number of seconds between updates of the data displayed in the table or chart.
- Use the **Download as** field to indicate whether to download a chart as a JPEG image or as a PNG image.
- Use **Colours selectors** to specify the colors to use on a chart.
- Set the **Show Acknowledged Alerts** switch to **Yes** if you want the table to display alerts that you acknowledged with a check box in the Ack'd column. Set it to **No** to hide any acknowledged alerts. Acknowledged alerts are purged from the table content only when the time specified in the alert definition passes.

To save your customizations, select **Save** (a checkmark) in the upper-right corner. To delete any previous changes and revert to the default values, select **Delete**. Use the **Save** and **Delete** menus to specify whether to apply your preferences to all dashboards or to a selected server or database.

Managing alerts

Use the PEM client's **Manage Alerts** tab to define, copy, or manage alerts. To open the **Manage Alerts** tab, select **Management > Manage Alerts**.

Quick Links

Copy Alerts

Alert Templates

Email Groups

Webhooks

Server Configuration

Help

Alerts

Manage Alerts

+

|                                     | Name   | Auto created?  | Template                             | Enable?        | Interval       |         | History retention |      |
|-------------------------------------|--|----------------|--------------------------------------|----------------|----------------|---------|-------------------|------|
|                                     |  |                |                                      |                | Default?       | Minutes | Default?          | Days |
| <input checked="" type="checkbox"/> | Average table bloat in server                  | <div>Yes</div> | Average table bloat in server        | <div>Yes</div> | <div>Yes</div> | 1       | <div>Yes</div>    | 30   |
| <input checked="" type="checkbox"/> | Connections in idle-in-transaction state       | <div>Yes</div> | Connections in idle-in-transactio... | <div>Yes</div> | <div>Yes</div> | 1       | <div>Yes</div>    | 30   |
| <input checked="" type="checkbox"/> | Connections in idle-in-transaction state, a... | <div>Yes</div> | Connections in idle-in-transactio... | <div>Yes</div> | <div>Yes</div> | 1       | <div>Yes</div>    | 30   |
| <input checked="" type="checkbox"/> | Connections in idle state                      | <div>Yes</div> | Connections in idle state            | <div>Yes</div> | <div>Yes</div> | 1       | <div>Yes</div>    | 30   |
| <input checked="" type="checkbox"/> | Database size in server                        | <div>Yes</div> | Database size in server              | <div>Yes</div> | <div>Yes</div> | 1       | <div>Yes</div>    | 30   |
| <input checked="" type="checkbox"/> | Highest table bloat in server                  | <div>Yes</div> | Highest table bloat in server        | <div>Yes</div> | <div>Yes</div> | 1       | <div>Yes</div>    | 30   |
| <input checked="" type="checkbox"/> | Largest index by table-size percentage         | <div>Yes</div> | Largest index by table-size perce... | <div>Yes</div> | <div>Yes</div> | 1       | <div>Yes</div>    | 30   |
| <input checked="" type="checkbox"/> | Last AutoVacuum                                | <div>Yes</div> | Last AutoVacuum                      | <div>Yes</div> | <div>Yes</div> | 1       | <div>Yes</div>    | 30   |
| <input checked="" type="checkbox"/> | Last Vacuum                                    | <div>Yes</div> | Last Vacuum                          | <div>Yes</div> | <div>Yes</div> | 1       | <div>Yes</div>    | 30   |

Use the Quick Links toolbar to open dialog boxes and tabs for managing alerts:

- Select **Copy Alerts** to open the Copy Alert Configuration dialog box and copy an alert definition.
- Select **Alert Templates** to open the **Alert Template** tab and modify or create an alert template.
- Select **Email Groups** to open the **Email Groups** tab and modify or create an email group.
- Select **Webhooks** to open the **Webhooks** tab and create or manage the webhooks endpoints.
- Select **Server Configurations** to open the Server Configuration dialog box and review or modify server configuration settings.
- Select **Help** to open the PEM online help in a new tab.

Use the table in the Alerts section of the **Manage Alerts** tab to create new alerts or manage existing alerts.

Alert templates

An alert template is a prototype that defines the properties of an alert. An alert instructs the server to compare the current state of the monitored object to a threshold specified in the alert template to determine if a situation requires administrative attention.

You can use the **Alert Templates** tab to define a custom alert template or view the definitions of existing alert templates. To open the **Alert Templates** tab, select **Management > Manage Alerts**. From the **Manage Alerts** tab, on the Quick Links toolbar, select **Alert Templates**.

Use the **Show System Template** list to filter the alert templates that are displayed in the Alert Templates table. From the list, select a level of the PEM hierarchy to view all of the templates for that level.

Defining a new alert template

To define a new alert template, from the **Show System Template** list, select **None**. Then click the plus sign (+) in the upper-right corner of the alert template table. The alert template editor opens.

Use fields on the **General** tab to specify general information about the template:



- Use the **Template name** field to specify a name for the new alert template.
- Use the **Description** field to provide a description of the alert template.
- Use the **Target type** list to select the type of object that is the focus of the alert.
- Use the **Applies to server** list to specify the server type (EDB Postgres Advanced Server or PostgreSQL) to which to apply the alert. You can specify a single server type or **ALL**.
- Use the **History retention** field to specify the number of days to store the result of the alert execution on the PEM server.
- Use the **Threshold unit** field to specify the unit type of the threshold value.
- Use fields in the **Auto create** box to specify for PEM to use the template to generate an automatic alert. If you enable this option, PEM creates an alert when a new server or agent, as specified by the **Target type** list, is added and deletes that alert when the target object is dropped.
  - Move the **Auto create?** slider to **Yes** to specify for PEM to create alerts based on the template. If you modify an existing alert template by changing the **Auto create?** slider to **Yes**, PEM creates alerts on the existing agents and servers. If you change the slider from **Yes** to **No**, the default threshold values in existing alerts are erased, and you can't recover them.
  - Use the **Operator** list to select the operator for PEM to use when evaluating the current system values.

Select a greater-than sign (>) to trigger the alert when the system values are greater than the values entered in the **Threshold values** fields.

Select a less-than sign (<) to indicate to trigger the alert when the system values are less than the values entered in the **Threshold values** fields.

- Use the threshold fields to specify the values for PEM to compare to the system values to determine whether to raise an alert. You must specify values for all three thresholds (Low, Medium, and High).
- Use the **Check frequency** field to specify the default number of minutes between alert executions. This value specifies how often the server invokes the SQL code specified in the definition and compares the result to the threshold value specified in the template.

Use the fields on the **Probe Dependency** tab to specify the names of probes referred to in the SQL query specified on the **SQL** tab:

- Use the **Probes** list to select from a list of the available probes.
  - To add the probe to the list of probes used by the alert template, select a probe name and select **Add**.
  - To remove a probe from the selected probes list, select the probe name and select **Delete**.
- Use the **Parameters** tab to define the parameters to use in the SQL code specified on the **SQL** tab. Select the plus sign (+). Then:
  - Use the **Name** field to specify the parameter name.
  - Use the **Data type** list to specify the type of parameter.
  - Use the **Unit** field to specify the type of unit specified by the parameter.
- Use the **Code** field on the **SQL** tab to provide the text of the SQL query for the server to invoke when executing the alert. The SQL query provides the result against which to compare the threshold value. If the alert result deviates from the specified threshold value, an alert is raised.

In the query, reference parameters defined on the **Parameters** tab sequentially by using the variable param\_**x**. The **x** indicates the position of the parameter definition in the parameter list. For example, param\_1 refers to the first parameter in the parameter list, param\_2 refers to the second parameter in the parameter list, and so on.

The query can also include the following variables:

| Variable description | Variable name |
|----------------------|---------------|
|----------------------|---------------|

| Variable description | Variable name       |
|----------------------|---------------------|
| agent identifier     | '\${agent_id}'      |
| server identifier    | '\${server_id}'     |
| database name        | '\${database_name}' |
| schema name          | '\${schema_name}'   |
| Table                | '\${object_name}'   |
| index                | '\${object_name}'   |
| sequence             | '\${object_name}'   |
| function name        | '\${object_name}'   |

- Use the **Detailed Information SQL** field to provide a SQL query to invoke if the alert is triggered. The result set of the query might be displayed as part of the detailed alert information on the Alerts dashboard or Global Overview dashboard.

#### Note

If the specified query depends on one or more probes from different levels in the PEM hierarchy (server, database, schema, and so on), and a probe becomes disabled, any resulting alerts are displayed as follows:

- If the alert definition and the probe referenced by the query are from the same level in the PEM hierarchy, the server displays any alerts that reference the alert template on the Alert Error table of the Global Alert dashboard.
- If the alert definition and the probe referenced by the query are from different levels of the PEM hierarchy, the server displays any triggered alerts that reference the alert template on the Alert Details table of the hierarchy on which the alert was defined.

To save the alert template definition and add the template name to the Alert Templates list, select **Save**. After saving a custom alert template, you can use the Alerting dialog box to define an alert based on the template.

## Exporting or importing alert templates

### New Feature

Export and import of the alert templates to another PEM Server is available from 8.2.0 and later.

To export the alert template:

1. Select any alert template from the **Alert Templates** tab.
2. Select **Export** in the upper-right corner of the table.
3. Select **Save File**.
4. To generate the JSON file, select **OK**.

To import the Alert Template:

1. On the **Alert Templates** tab, select **Import** in the upper-right corner.
2. To select the JSON file with the code import, select **Browse**, and then select **Import**.
3. After selecting the file to import, you can select the following check boxes:
  - **Skip existing** — Skip the alert template if it already exists.
  - **Skip existing dependent probe** — The alert templates depend on probes. Select this check box to skip the dependent probe if it already exists.

If both the check boxes are selected and the alert template already exists, then it skips importing the alert template.

If you don't select the **Skip existing** check box, select **Skip dependent probe**, and the alert template already exists, then the alert template imports

successfully.

If both the check boxes are cleared and the alert template doesn't exist, then it successfully imports the alert template.

Modifying or deleting an alert template

To view the definition of an existing template (including PEM predefined alert templates), use the **Show System Template** list to select the type of object monitored. When you select the object type, the Alert Templates table displays the alert templates that correspond with that object type.

Select a template name in the list, and select **Edit** at the left end of the row to review the template definition.

Use the Alert Templates dialog box to view detailed information about the alert template:

- The **General** tab displays general information.
- The **Probe Dependency** tab lists the names of probes that provide data for the template.
- The **Parameters** tab lists the names of any parameters referred to in the SQL code.
- The **SQL** tab displays the SQL code that defines the behavior of the alert.

To delete an alert template, select the template name in the alert templates table and select **Delete**, located in the upper-right corner of the table. The alert history persists for the time specified in the **History Retention** field in the template definition.

Predefined alert templates – reference

An alert definition contains a system-defined or user-defined set of conditions that PEM compares to the system statistics. If the statistics deviate from the boundaries specified for that statistic, the alert triggers, and the PEM client displays a warning on the Alerts Overview page and optionally sends a notification to a monitoring user.

The tables that follow list the system-defined alert templates that you can use to create an alert. This list is subject to change and can vary by system.

Templates applicable on agent

| Template name                          | Description   |
|--|---|
| Load Average (1 minute)                | 1-minute system load average                                    |
| Load Average (5 minutes)               | 5-minute system load average                                    |
| Load Average (15 minutes)              | 15-minute system load average                                   |
| Load Average per CPU Core (1 minutes)  | 1-minute system load average per CPU core                       |
| Load Average per CPU Core (5 minutes)  | 5-minute system load average per CPU core                       |
| Load Average per CPU Core (15 minutes) | 15-minute system load average per CPU core                      |
| CPU utilization                        | Average CPU consumption   |
| Number of CPUs running higher than a   | Number of CPUs running at greater than K% utilization threshold |
| Free memory percentage                 | Free memory as a percent of total system memory                 |
| Memory used percentage                 | Percentage of memory used                                       |
| Swap consumption                       | Swap space consumed (in megabytes)                              |
| Swap consumption percentage            | Percentage of swap area consumed                                |
| Disk Consumption                       | Disk space consumed (in megabytes)                              |
| Disk consumption percentage            | Percentage of disk consumed                                     |
| Disk Available                         | Disk space available (in megabytes)                             |

| Template name  | Description  |
|--|--|
| Disk busy percentage   | Percentage of disk busy  |
| Most used disk percentage  | Percentage used of the most utilized disk on the system                    |
| Total table bloat on host  | The total space wasted by tables on a host, in MB                          |
| Highest table bloat on host  | The most space wasted by a table on a host, in MB                          |
| Average table bloat on host  | The average space wasted by tables on host, in MB                          |
| Table size on host   | The size of tables on host, in MB  |
| Database size on host  | The size of databases on host, in MB                                       |
| Number of ERRORS in the logfile on agent N in last X hours.            | The number of ERRORS in the logfile on agent N in last X hours             |
| Number of WARNINGS in the logfile on agent N in last X hours           | The number of WARNINGS in the logfile on agent N in last X hours           |
| Number of WARNINGS or ERRORS in the logfile on agent N in last X hours | The number of WARNINGS or ERRORS in the logfile on agent N in last X hours |
| Package version mismatch   | Check for package version mismatch as per catalog                          |
| Total materialized view bloat on host                                  | The total space wasted by materialized views on a host, in MB              |
| Highest materialized view bloat on host                                | The most space wasted by a materialized view on a host, in MB              |
| Average materialized view bloat on host                                | The average space wasted by materialized views on host, in MB              |
| Materialized view size on host   | The size of materialized views on host, in MB                              |
| Agent Down   | Specified agent is currently down  |

#### Templates applicable on server

| Template name  | Description   |
|--|---|
| Total table bloat in server  | The total space wasted by tables in server, in MB   |
| Largest table (by multiple of unbloated size)                              | Largest table in server, calculated as a multiple of its own estimated unbloated size; exclude tables smaller than N MB                   |
| Highest table bloat in server  | The most space wasted by a table in server, in MB   |
| Average table bloat in server  | The average space wasted by tables in server, in MB   |
| Table size in server   | The size of tables in server, in MB   |
| Database size in server  | The size of databases in server, in MB  |
| Number of WAL files  | Total number of Write Ahead Log files   |
| Number of prepared transactions  | Number of transactions in prepared state  |
| Total connections  | Total number of connections in the server   |
| Total connections as percentage of   | Total number of connections in the server as a percentage of maximum  |
| max_connections  | connections allowed on server, settings   |
| Unused, non-superuser connections  | Number of unused, non-superuser connections on the server, user_info, settings  |
| Unused, non-superuser connections as percentage of max_connections         | Number of unused, non-superuser connections on the server as a percentage of max_connections of max_connections, user_info, settings      |
| Ungranted locks  | Number of ungranted locks in server   |
| Percentage of buffers written by backends                                  | The percentage of buffers written by backends vs. the total buffers written   |
| Percentage of buffers written by checkpoint                                | The percentage of buffers written by the checkpoints vs. the total buffers written  |
| Buffers written per second   | Number of buffers written per second, over the last two probe cycles  |
| Buffers allocated per second   | Number of buffers allocated per second, over the last two probe cycles  |
| Connections in idle state  | Number of connections in server that are in idle state  |
| Connections in idle-in-transaction state                                   | Number of connections in server that are in idle-in-transaction state   |
| Connections in idle-in-transaction state, as percentage of max_connections | Number of connections in server that are in idle-in-transaction state, as a percentage of maximum connections allowed on server, settings |

| Template name   | Description   |
|---|---|
| Long-running idle connections   | Number of connections in the server that have been idle for more than N seconds                                     |
| Long-running idle connections and idle transactions                         | Number of connections in the server that have been idle or transactions idle-in-transaction for more than N seconds |
| Long-running idle transactions  | Number of connections in the server that have been idle in transaction for more than N seconds                      |
| Long-running transactions   | Number of transactions in server that have been running for more than N seconds                                     |
| Long-running queries  | Number of queries in server that have been running for more than N seconds  |
| Long-running vacuums  | Number of vacuum operations in server that have been running for more than N seconds                                |
| Long-running autovacuums  | Number of autovacuum operations in server that have been running for more than N seconds                            |
| Committed transactions percentage   | Percentage of transactions in the server that committed vs. that rolled-back over last N minutes                    |
| Shared buffers hit percentage   | Percentage of block read requests in the server that were satisfied by shared buffers, over last N minutes          |
| Tuples inserted   | Tuples inserted into server over last N minutes   |
| InfiniteCache buffers hit percentage  | Percentage of block read requests in the server that were satisfied by InfiniteCache, over last N minutes           |
| Tuples fetched  | Tuples fetched from server over last N minutes  |
| Tuples returned   | Tuples returned from server over last N minutes   |
| Dead Tuples   | Number of estimated dead tuples in server   |
| Tuples updated  | Tuples updated in server over last N minutes  |
| Tuples deleted  | Tuples deleted from server over last N minutes  |
| Tuples hot updated  | Tuples hot updated in server, over last N minutes   |
| Sequential Scans  | Number of full table scans in server, over last N minutes   |
| Index Scans   | Number of index scans in server, over last N minutes  |
| Hot update percentage   | Percentage of hot updates in the server over last N minutes   |
| Live Tuples   | Number of estimated live tuples in server   |
| Dead tuples percentage  | Percentage of estimated dead tuples in server   |
| Last Vacuum   | Hours since last vacuum on the server   |
| Last AutoVacuum   | Hours since last autovacuum on the server   |
| Last Analyze  | Hours since last analyze on the server  |
| Last AutoAnalyze  | Hours since last autoanalyze on the server  |
| Percentage of buffers written by backends over the last N minutes           | The percentage of buffers written by backends vs. the total buffers backends over last N                            |
| Table Count   | Total number of tables in server  |
| Function Count  | Total number of functions in server   |
| Sequence Count  | Total number of sequences in server   |
| A user expires in N days  | Number of days before a user's validity expires   |
| Index size as a percentage of table size                                    | Size of the indexes in server, as a percentage of their tables' size  |
| Largest index by table-size percentage oc_index, table_size                 | Largest index in server, calculated as percentage of its table's size   |
| Number of ERRORS in the logfile on server M in the last X hours             | The number of ERRORS in the logfile on server M in last X hours   |
| Number of WARNINGS in the logfile on server M in the last X hours           | The number of WARNINGS in logfile on server M in the last X hours   |
| Number of WARNINGS or ERRORS in the logfile on server M in the last X hours | The number of WARNINGS or ERRORS in the logfile on server M in the last X hours                                     |
| Number of attacks detected in the last N minutes                            | The number of SQL injection attacks occurred in the last N minutes  |
| Number of attacks detected in the last N minutes by username                | The number of SQL injection attacks occurred in the last N minutes by username                                      |

| Template name   | Description   |
|---|---|
| Number of replica servers lag behind the primary by write location  | Streaming Replication: number of replica servers lag behind the primary by write location   |
| Number of replica servers lag behind the primary by flush location  | Streaming Replication: number of replica servers lag behind the primary by flush location   |
| Number of replica servers lag behind the primary by replay location | Streaming Replication: number of replica servers lag behind the primary by replay location  |
| Replica server lag behind the primary by write location             | Streaming Replication: replica server lag behind the primary by write location in MB  |
| Replica server lag behind the primary by flush location             | Streaming Replication: replica server lag behind the primary by flush location in MB  |
| Replica server lag behind the primary by replay location            | Streaming Replication: replica server lag behind the primary by replay location in MB   |
| Replica server lag behind the primary by size (MB)                  | Streaming Replication: replica server lag behind the primary by size in MB  |
| Replica server lag behind the primary by WAL segments               | Streaming Replication: replica server lag behind the primary by WAL segments  |
| Replica server lag behind the primary by WAL pages                  | Streaming Replication: replica server lag behind the primary by WAL pages   |
| Total materialized view bloat in server                             | The total space wasted by materialized views in server, in MB   |
| Largest materialized view (by multiple of unbloated size)           | Largest materialized view in server, calculated as a multiple of its own estimated unbloated size; exclude materialized views smaller than N MB |
| Highest materialized view bloat in server                           | The most space wasted by a materialized view in server, in MB   |
| Average materialized view bloat in server                           | The average space wasted by materialized views in server, in MB   |
| Materialized view size in server                                    | The size of materialized view in server, in MB  |
| View Count  | Total number of views in server   |
| Materialized View Count   | Total number of materialized views in server  |
| Audit config mismatch   | Check for audit config parameter mismatch   |
| Server Down   | Specified server is currently inaccessible  |
| Number of WAL archives pending                                      | Streaming Replication: number of WAL files pending to be replayed at replica  |
| Number of minutes lag of replica server from primary server         | Streaming Replication: number of minutes replica node is lagging behind the primary node  |
| Log config mismatch   | Check for log config parameter mismatch   |

#### Templates applicable on database

| Template name                                      | Description   |
|--|---|
| Total table bloat in database                      | The total space wasted by tables in database, in MB   |
| Largest table (by multiple of unbloated size)      | Largest table in database, calculated as a multiple of its own estimated unbloated size; exclude tables smaller than N MB |
| Highest table bloat in database                    | The most space wasted by a table in database, in MB   |
| Average table bloat in database                    | The average space wasted by tables in database, in MB   |
| Table size in database                             | The size of tables in database, in MB   |
| Database size                                      | The size of the database, in MB   |
| Total connections                                  | Total number of connections in the database   |
| Total connections as percentage of max_connections | Total number of connections in the database as a percentage of maximum connections allowed on server, settings            |
| Ungranted locks                                    | Number of ungranted locks in database   |
| Connections in idle state                          | Number of connections in database that are in idle state  |
| Connections in idle-in-transaction state           | Number of connections in database that are in idle-in-transaction state   |

| Template name   | Description   |
|---|---|
| Connections in idle-in-transaction state,as percentage of max_connections | Number of connections in database that are in idle-in-transaction state, as a percentage of maximum connections allowed on server, settings |
| Long-running idle connections   | Number of connections in the database that have been idle for more than N seconds   |
| Long-running idle connections and idle transactions                       | Number of connections in the database that have been idle or idle-in-transaction for more than N seconds                                    |
| Long-running idle transactions  | Number of connections in the database that have been idle in transaction for more than N seconds  |
| Long-running transactions   | Number of transactions in database that have been running for more than N seconds   |
| Long-running queries  | Number of queries in database that have been running for more than N seconds  |
| Long-running vacuums  | Number of vacuum operations in database that have been running for more than N seconds  |
| Long-running autovacuums  | Number of autovacuum operations in database that have been running for more than N seconds  |
| Committed transactions percentage   | Percentage of transactions in the database that committed vs. that rolled-back over last N minutes  |
| Shared buffers hit percentage   | Percentage of block read requests in the database that were satisfied by shared buffers, over last N minutes                                |
| InfiniteCache buffers hit percentage                                      | Percentage of block read requests in the database that were satisfied by InfiniteCache, over last N minutes                                 |
| Tuples fetched  | Tuples fetched from database over last N minutes  |
| Tuples returned   | Tuples returned from database over last N minutes   |
| Tuples inserted   | Tuples inserted into database over last N minutes   |
| Tuples updated  | Tuples updated in database over last N minutes  |
| Tuples deleted  | Tuples deleted from database over last N minutes  |
| Tuples hot updated  | Tuples hot updated in database, over last N minutes   |
| Sequential Scans  | Number of full table scans in database, over last N minutes   |
| Index Scans   | Number of index scans in database, over last N minutes  |
| Hot update percentage   | Percentage of hot updates in the database over last N minutes   |
| Live Tuples   | Number of estimated live tuples in database   |
| Dead Tuples   | Number of estimated dead tuples in database   |
| Dead tuples percentage  | Percentage of estimated dead tuples in database   |
| Last Vacuum   | Hours since last vacuum on the database   |
| Last AutoVacuum   | Hours since last autovacuum on the database   |
| Last Analyze  | Hours since last analyze on the database  |
| Last AutoAnalyze  | Hours since last autoanalyze on the database  |
| Table Count   | Total number of tables in database  |
| Function Count  | Total number of functions in database   |
| Sequence Count  | Total number of sequences in database   |
| Index size as a percentage of table size                                  | Size of the indexes in database, as a percentage of their tables' size  |
| Largest index by table-size percentage                                    | Largest index in database, calculated as percentage of its table's size, oc_index, table_size   |
| Database Frozen XID   | The age (in transactions before the current transaction) of the database's frozen transaction ID  |
| Number of attacks detected in the   | The number of SQL injection attacks occurred in the last N minutes  |
| Number of attacks detected in the   | The number of SQL injection attacks occurred in the last N minutes by last N minutes by username  |
| Queries that have been cancelled due to dropped tablespaces               | Streaming Replication: number of queries that have been cancelled due to dropped tablespaces  |
| Queries that have been cancelled due to lock timeouts                     | Streaming Replication: number of queries that have been cancelled due to lock timeouts  |

| Template name   | Description   |
|---|---|
| Queries that have been cancelled due to old snapshots     | Streaming Replication: number of queries that have been cancelled due to old snapshots  |
| Queries that have been cancelled due to pinned buffers    | Streaming Replication: number of queries that have been cancelled due to pinned buffers   |
| Queries that have been cancelled due to deadlocks         | Streaming Replication: number of queries that have been cancelled due to deadlocks  |
| Total events lagging in all slony clusters                | Slony Replication: total events lagging in all slony clusters   |
| Events lagging in one slony cluster                       | Slony Replication: events lagging in one slony cluster  |
| Lag time (minutes) in one slony cluster                   | Slony Replication: lag time (minutes) in one slony cluster  |
| Total rows lagging in xdb single primary replication      | xDB Replication: Total rows lagging in xdb single primary replication   |
| Total rows lagging in xdb multi primary replication       | xDB Replication: Total rows lagging in xdb multi primary replication  |
| Total materialized view bloat in database                 | The total space wasted by materialized views in database, in MB   |
| Largest materialized view (by multiple of unbloated size) | Largest materialized view in database, calculated as a multiple of its estimated unbloated size; exclude materialized views smaller than N MB |
| Highest materialized view bloat in database               | The most space wasted by a materialized view in database, in MB   |
| Average materialized view bloat in database               | The average space wasted by materialized views in database, in MB   |
| Materialized view size in database                        | The size of materialized view in database, in MB  |
| View Count  | Total number of views in database   |
| Materialized View Count                                   | Total number of materialized views in database  |

#### Templates applicable on schema

| Template name                                 | Description   |
|---|---|
| Total table bloat in schema                   | The total space wasted by tables in schema, in MB   |
| Largest table (by multiple of unbloated size) | Largest table in schema, calculated as a multiple of its own estimated unbloated size; exclude tables smaller than N MB |
| Highest table bloat in schema                 | The most space wasted by a table in schema, in MB   |
| Average table bloat in schema                 | The average space wasted by tables in schema, in MB   |
| Table size in schema                          | The size of tables in schema, in MB   |
| Tuples inserted                               | Tuples inserted in schema over last N minutes   |
| Tuples updated                                | Tuples updated in schema over last N minutes  |
| Tuples deleted                                | Tuples deleted from schema over last N minutes  |
| Tuples hot updated                            | Tuples hot updated in schema, over last N minutes   |
| Sequential Scans                              | Number of full table scans in schema, over last N minutes   |
| Index Scans                                   | Number of index scans in schema, over last N minutes  |
| Hot update percentage                         | Percentage of hot updates in the schema over last N minutes   |
| Live Tuples                                   | Number of estimated live tuples in schema   |
| Dead Tuples                                   | Number of estimated dead tuples in schema   |
| Dead tuples percentage                        | Percentage of estimated dead tuples in schema   |
| Last Vacuum                                   | Hours since last vacuum on the schema   |
| Last AutoVacuum                               | Hours since last autovacuum on the schema   |
| Last Analyze                                  | Hours since last analyze on the schema  |
| Last AutoAnalyze                              | Hours since last autoanalyze on the schema  |
| Table Count                                   | Total number of tables in schema  |
| Function Count                                | Total number of functions in schema   |
| Sequence Count                                | Total number of sequences in schema   |



| Template name   | Description  |
|---|--|
| Index size as a percentage of table size                  | Size of the indexes in schema, as a percentage of their table's size   |
| Largest index by table-size percentage                    | Largest index in schema, calculated as percentage of its table's size, oc_index, table_size  |
| Materialized View bloat                                   | Space wasted by the materialized view, in MB   |
| Total materialized view bloat in schema                   | The total space wasted by materialized views in schema, in MB  |
| Materialized view size as a multiple of unbloated size    | Size of the materialized view as a multiple of estimated unbloated size  |
| Largest materialized view (by multiple of unbloated size) | Largest materialized view in schema, calculated as a multiple of its own estimated unbloated size; exclude materialized view smaller than N MB |
| Highest materialized view bloat in schema                 | The most space wasted by a materialized view in schema, in MB  |
| Average materialized view bloat in schema                 | The average space wasted by materialized views in schema, in MB  |
| Materialized view size                                    | The size of materialized view, in MB   |
| Materialized view size in schema                          | The size of materialized views in schema, in MB  |
| View Count  | Total number of views in schema  |
| Materialized View Count                                   | Total number of materialized views in schema   |
| Materialized View Frozen XID                              | The age (in transactions before the current transaction) of the materialized view's frozen transaction ID                                      |

Templates applicable on table

| Template name                             | Description   |
|---|---|
| Table bloat                               | Space wasted by the table, in MB  |
| Table size                                | The size of table, in MB  |
| Table size as a multiple of ubloated size | Size of the table as a multiple of estimated unbloated size                                   |
| Tuples inserted                           | Tuples inserted in table over last N minutes  |
| Tuples updated                            | Tuples updated in table over last N minutes   |
| Tuples deleted                            | Tuples deleted from table over last N minutes   |
| Tuples hot updated                        | Tuples hot updated in table, over last N minutes  |
| Sequential Scans                          | Number of full table scans on table, over last N minutes                                      |
| Index Scans                               | Number of index scans on table, over last N minutes   |
| Hot update percentage                     | Percentage of hot updates in the table over last N minutes                                    |
| Live Tuples                               | Number of estimated live tuples in table  |
| Dead Tuples                               | Number of estimated dead tuples in table  |
| Dead tuples percentage                    | Percentage of estimated dead tuples in table  |
| Last Vacuum                               | Hours since last vacuum on the table  |
| Last AutoVacuum                           | Hours since last autovacuum on the table  |
| Last Analyze                              | Hours since last analyze on the table   |
| Last AutoAnalyze                          | Hours since last autoanalyze on the table   |
| Row Count                                 | Estimated number of rows in a table   |
| Index size as a percentage of table size  | Size of the indexes on table, as a percentage of table's size                                 |
| Table Frozen XID                          | The age (in transactions before the current transaction) of the table's frozen transaction ID |

Global templates

| Template name | Description  |
|---------------|--|
| Agents Down   | Number of agents that haven't reported in recently |
| Servers Down  | Number of servers that are currently inaccessible  |
| Alert Errors  | Number of alerts in an error state                 |

#### Audit log alerting

PEM provides alert templates that let you use the Alerting dialog to create an alert that triggers when an **ERROR** or **WARNING** statement is written to a log file for a specific server or agent. To open the Alerting dialog, select the server or agent in the PEM client Object browser tree control, and select **Management > Alerting**.

To create an alert to notify you of error or warning messages in the log file for a specific server, create an alert that uses one of the following alert templates:

- Number of ERRORS in the logfile on server M in last X hours
- Number of WARNINGS in the logfile on server M in last X hours
- Number of ERRORS or WARNINGS in the logfile on server M in last X hours

To create an alert to notify you of error or warning messages for a specific agent, create an alert that uses one of the following alert templates. This functionality is supported only on EDB Postgres Advanced Server.


- Number of ERRORS in the logfile on agent M in last X hours
- Number of WARNINGS in the logfile on agent M in last X hours
- Number of ERRORS or WARNINGS in the logfile on agent M in last X hours


#### Defining a new alert


Use the PEM client **Manage Alerts** tab to define, copy, or manage alerts. To open the **Manage Alerts** tab, select **Management > Manage Alerts**.


The **Manage Alerts** tab displays a table of alerts that are defined on the object currently selected in the PEM client tree. You can use the Alerts table to modify an existing alert or to create a new alert.


Quick Links


Copy Alerts

Alert Templates

Email Groups










Webhooks

Server Configuration

Help

Alerts

Manage Alerts

|                                     |   | Name   | Auto created?                       | Template                             | Enable?                             | Interval                            |         | History retention |                                     |    |
|-------------------------------------|---|--|-------------------------------------|--------------------------------------|-------------------------------------|-------------------------------------|---------|-------------------|-------------------------------------|----|
|                                     |   |  |                                     |                                      |                                     | Default?                            | Minutes | Default?          | Days                                |    |
| <input checked="" type="checkbox"/> |  | Average table bloat in server                  | <input checked="" type="checkbox"/> | Average table bloat in server        | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |         | 1                 | <input checked="" type="checkbox"/> | 30 |
| <input checked="" type="checkbox"/> |  | Connections in idle-in-transaction state       | <input checked="" type="checkbox"/> | Connections in idle-in-transactio... | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |         | 1                 | <input checked="" type="checkbox"/> | 30 |
| <input checked="" type="checkbox"/> |  | Connections in idle-in-transaction state, a... | <input checked="" type="checkbox"/> | Connections in idle-in-transactio... | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |         | 1                 | <input checked="" type="checkbox"/> | 30 |
| <input checked="" type="checkbox"/> |  | Connections in idle state                      | <input checked="" type="checkbox"/> | Connections in idle state            | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |         | 1                 | <input checked="" type="checkbox"/> | 30 |
| <input checked="" type="checkbox"/> |  | Database size in server                        | <input checked="" type="checkbox"/> | Database size in server              | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |         | 1                 | <input checked="" type="checkbox"/> | 30 |
| <input checked="" type="checkbox"/> |  | Highest table bloat in server                  | <input checked="" type="checkbox"/> | Highest table bloat in server        | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |         | 1                 | <input checked="" type="checkbox"/> | 30 |
| <input checked="" type="checkbox"/> |  | Largest index by table-size percentage         | <input checked="" type="checkbox"/> | Largest index by table-size perce... | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |         | 1                 | <input checked="" type="checkbox"/> | 30 |
| <input checked="" type="checkbox"/> |  | Last AutoVacuum                                | <input checked="" type="checkbox"/> | Last AutoVacuum                      | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |         | 1                 | <input checked="" type="checkbox"/> | 30 |
| <input checked="" type="checkbox"/> |  | Last Vacuum                                    | <input checked="" type="checkbox"/> | Last Vacuum                          | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |         | 1                 | <input checked="" type="checkbox"/> | 30 |

To open the alert editor and create an alert, select the plus sign (+) in the upper-right of the table. The editor opens.

Use the fields on the **General** tab to provide information about the alert:

- Enter the name of the alert in the **Name** field.
- Use the **Template** list to select a template for the alert. An alert template is a function that uses one or more metrics or parameters to generate a value to which PEM compares user-specified alert boundaries. If the value returned by the template function evaluates to a value that's within the boundary of a user-defined alert as specified by the **Operator** and **Threshold values** fields, PEM:
  - Raises an alert
  - Adds a notice to the Alerts overview display
  - Performs any actions specified on the template
- Use the **Enable?** switch to specify if the alert is enabled (**Yes**) or disabled (**No**).
- Use the **Interval** box to specify how often the alert confirms if the alert conditions are satisfied. Use the **Minutes** selector to specify an interval value. Use the **Default** switch to set or reset the **Minutes** value to the default (recommended) value for the selected template.
- Use the **History retention** box to specify the number of days that PEM stores data collected by the alert. Use the **Days** selector to specify the number of days to store the data. Use the **Default** switch to set or reset the **Days** value to the default value (30 days).
- Use controls in the **Threshold values** box to define the triggering criteria for the alert. When the value specified in the **Threshold values** fields evaluates to greater than or less than the system value (as specified with the Operator), PEM raises a Low, Medium or High alert level.
- Use the **Operator** list to select the operator for PEM to use when evaluating the current system values:
  - Select a greater-than sign (>) to trigger the alert when the system values are greater than the values entered in the **Threshold values** fields.
  - Select a less-than sign (<) to trigger the alert when the system values are less than the values entered in the **Threshold values** fields.
- Use the **Threshold** fields to specify the values for PEM to compare to the system values to determine whether to raise an alert. You must specify values for all three thresholds (Low, Medium, and High).

The Parameter Options table contains a list of parameters that are required by the selected template. The table displays both predefined parameters and parameters for which you must specify a value. You must specify a value for any parameter that displays a prompt in the **Value** column.

PEM can send a notification or execute a script if an alert is triggered or if an alert is cleared. Use the **Notification** tab to specify how PEM behaves if an alert is raised.

Use the **Email notification** box to specify the email group to receive an email notification if the alert is triggered at the specified level. Use the [Email Groups](#) tab to create an email group that contains the address of the users to notify when an alert is triggered. To access the **Email Groups** tab, select **Email Groups** located in the **Quick Links** menu of the **Manage Alerts** tab.

- To instruct PEM to send an email when a specific alert level is reached, set the slider next to an alert level to **Yes**. Use the list to select the predefined user or group to notify.

You must configure the PEM server to use an SMTP server to deliver email before PEM can send email notifications.

Use the **Webhook notification** box to specify one or multiple endpoints if the alert is triggered at the specified level. Use the [webhooks tab](#) to create a webhook endpoint to receive the notifications when an alert is triggered. To access the **Webhooks** tab, select **Webhooks** located in the **Quick Links** menu of the **Manage Alerts** tab.

- Set **Enable?** to **Yes** to send the alert notifications to the webhook endpoint.
- Set **Override default configuration?** to **Yes** to set the customized alert levels as per the requirement. Once it's set to **Yes**, all the alert levels are enabled to configure.
- Use the list to select a predefined endpoint to send a notification to for **Low alerts?**, **Medium alerts?**, **High alerts?**, and **Cleared alerts?**.

Use the **Trap notification** options to configure trap notifications for this alert:

- Set **Send trap** to **Yes** to send SNMP trap notifications when the state of this alert changes.
- Set **SNMP Ver** to **v1**, **v2**, or **v3** to identify the SNMP version.
- Use the **Low alert**, **Med alert**, and **High alert** sliders to select the levels of alert to trigger the trap. For example, if you set the slider next to **High alert** to **Yes**, PEM sends a notification when an alert with a high-severity level is triggered.

You must configure the PEM server to send notifications to an SNMP trap/notification receiver before notifications can be sent. For sending SNMP v3 traps, pemAgent uses 'User Security Model(USM)', which is in charge of authenticating, encrypting, and decrypting SNMP packets.

While sending SNMP v3 traps, the agent creates the `snmp_boot_counter` file. This file is created in the location mentioned by the `batch_script_dir` parameter in `agent.cfg`. If this parameter isn't configured or if the directory isn't accessible due to authentication restrictions, then the file is created in the operating system temporary directory. If that's also not possible, then the file is created in your home directory.

Use the **Nagios notification** box to instruct the PEM server to notify Nagios network-alerting software when the alert is triggered or cleared. For more details, see [Using PEM with Nagios](#)

- Set the **Submit passive service check result to Nagios** switch to **Yes** to notify Nagios when the alert is triggered or cleared.
- Use the **Script execution** box to optionally define a script that executes if an alert is triggered and to specify details about the script execution.
- Set the **Execute script** slider to **Yes** to instruct PEM to execute the provided script if an alert is triggered.
- Set the **Execute on alert cleared** slider to **Yes** to instruct PEM to execute the provided script when the situation that triggered the alert is resolved.
- Use the **Execute script on** options to indicate for the script to execute on the PEM server or the monitored server.
- In the **Code** field, provide the script for PEM to execute. You can provide a batch/shell script or SQL code. In the script, you can use placeholders for the following:

`%AlertName%` — The name of the triggered alert.

`%ObjectName%` — The name of the server or agent on which the alert was triggered.

`%ThresholdValue%` — The threshold value reached by the metric when the alert triggered.

`%CurrentValue%` — The current value of the metric that triggered the alert.

`%CurrentState%` — The current state of the alert.

`%OldState%` — The previous state of the alert.

`%AlertRaisedTime%` — The time that the alert was raised or the most recent time that the alert state was changed.

To invoke a script on a Linux system, you must modify the entry for the `batch_script_user` parameter of the `agent.cfg` file and specify the user to use to run the script. You can either specify a non-root user or root for this parameter. If you don't specify a user or the specified user doesn't

exist, then the script doesn't execute. Restart the agent after modifying the file.

To invoke a script on a Windows system, set the registry entry for `AllowBatchJobSteps` to true and restart the PEM agent. PEM registry entries are located in `HKEY_LOCAL_MACHINE\Software\EnterpriseDB\PEM\agent`.

After you define the alert attributes, select **Edit** to close the alert definition editor and then **Save** in the upper-right corner of the Alerts table.

To discard your changes, select **Refresh**. A message prompts you to confirm that you want to discard the changes.

#### Note

Suppose you need to use the alert configuration placeholder values in an external script. You can do so either by passing them as the command-line arguments or exporting them as environment variables. The external script must have proper execution permissions.

- You can run the script with any of the placeholders as command-line arguments.

For example:

```
#!/bin/bash

bash <path_to_script>/script.sh "%AlertName% %AlertLevel% %AlertDetails%"
```

- You can define the environment variables for any of the placeholders and then use those environment variables in the script.

For example:

```
#!/bin/bash

export AlertName=%AlertName%
export AlertState=%AlertState%

bash <path_to_script>/script.sh
```

## Modifying an alert

Use the Alerts table to manage an existing alert or create a new alert. Select an object in the PEM client tree to view the alerts that monitor that object.

You can modify some properties of an alert in the Alerts table:

- The Alert name column displays the name of the alert. To change the alert name, replace the name in the table and select **Save**.
- The Alert template column displays the name of the alert template that specifies properties used by the alert. You can use the list to change the alert template associated with an alert.
- Use the **Alert enable?** switch to specify if an alert is enabled (**Yes**) or disabled (**No**).
- Use the Interval column to specify how often PEM checks whether the alert conditions are satisfied. Set the **Default** switch to **No** and specify an alternate value, in minutes. Or set the **Default** switch to **Yes** to reset the value to its default setting. By default, PEM checks the status of each alert once every minute.
- Use the **History retention** field to specify the number of days that PEM stores data collected by the alert. Set the **Default** switch to **No** and specify an alternative value in days. Or set the **Default** switch to **Yes** to reset the value to its default setting. By default, PEM stores historical data for 30 days.

After modifying an alert, select **Save** (located in the upper-right corner of the table) to preserve your changes.

To modify other alert attributes, select **Edit** to the left of an alert name to open an editor. The editor provides access to the complete alert definition.

Use the Alert Details dialog box to modify the definition of the selected alert. After you modify the alert definition, select **Save**.

## Deleting an alert

To mark an alert for deletion, select the alert name in the Alerts table. Then select **Delete** to the left of the name. The alert remains in the list in red strike-through font.

**Delete** is a toggle. You can undo the deletion by selecting it a second time. To permanently delete the alert definition, select **Save**.

## Copying an alert

To speed up the deployment of alerts in the PEM system, you can copy alert definitions from one object to one or more target objects.

To copy alerts from an object, select the object in the PEM client tree on the main PEM window. Then, select **Management > Copy Alerts**. On the **Manage Alerts** tab, from the Quick Links toolbar, select **Copy Alerts**.

The Copy Alert Configuration dialog box copies all alerts from the object selected in the PEM client tree to the objects selected on the dialog box. Expand the tree to select nodes to specify as the target objects. The tree displays a red warning indicator next to the source object.

To copy alerts to multiple objects at once, select a parent node of the targets. For example, to copy the alerts from one table to all tables in a schema, select the check box next to the schema. PEM copies alerts only to targets that are the same type as the source object.

Select **Ignore duplicates** to prevent PEM from updating any existing alerts on the target objects with the same name as those being copied. Select **Replace duplicates** to replace existing alerts with alerts of the same name from the source object.

Select **Configure Alerts** to copy the alerts from the source object to all objects of the same type in or under those objects selected on the Copy Alert Configuration dialog box.

## Schedule an alert blackout

You can use the **Management > Schedule Alert Blackout** to schedule an alert blackout for your Postgres servers and PEM agents during maintenance. Alerts aren't raised during a defined blackout period.

To schedule an alert blackout, select **Management > Schedule Alert Blackout**.

In the Schedule Alert Blackout dialog box, use the tabs to define the blackout period for servers and agents. On the **Server** tab, to add a row, select the plus sign (+) at the top-right corner.

Use the **Server** tab to provide information about an alert blackout period. After you save the blackout period, you can't edit it.

- Use the **Start time** field to provide the date and time to start the alert blackout.
- Use the **Duration** field to provide the interval for which you want to black out the alerts.
- Use the **Servers** field to provide the server name for which you want to black out the alerts. You can also select multiple servers to black out the alerts for all of those servers.

After providing details, select **Save**. The alerts don't appear on the Alerts dashboard for the scheduled interval of that server.

You can also schedule a blackout period for PEM agents using the **Agent** tab on the dialog box. To add a row, on the **Agent** tab, select the plus sign (+) at the top-right corner.

Use the **Agent** tab to provide the information about an alert blackout period. After you save the blackout period, you can't edit it.

- Use the **Start time** field to provide the date and time to start the alert blackout.
- Use the **Duration** field to provide the interval for which you want to black out the alerts.
- Use the **Agents** field to provide the agent name for which you want to black out the alerts. All server-level alerts for the servers bound to that agent

black out.

After providing details, save the details by selecting **Save**. The alerts aren't displayed on the Alert dashboard for the scheduled interval for that PEM agent.

You can select **Clone** from the top-right corner of the dialog box to clone the scheduling of an alert blackout. To create the cloned copy of all the selected servers or agents, select the servers or agents you want to clone, and then select **Clone**. You can edit newly created schedules as needed, and then select **Save**.

Select **Delete** from the top-right corner of the dialog box to remove a scheduled alert blackout. Select the servers or agents and then select **Delete**.

Select a server for which you want to delete the scheduled alert backout, and then select **Delete**. The server prompts for confirmation before deleting that row.

You can select **Reset** to reset the details on the Alert Blackout dialog box to the default settings. Saved blackouts aren't affected.

## 22.3 Notifications

PEM can send a notification or execute a script if an alert is triggered or cleared. You can send notifications using the following options:

- SMTP
- Webhooks
- SNMP
- Nagios

Use the **Notification** tab to specify PEM behavior when an alert is raised.

### SMTP

You must configure the PEM server to use an SMTP server to deliver email before PEM can send email notifications.

#### Creating an email group

PEM monitors your system for conditions that require attention. You can use an email group to specify the email addresses of users for the server to notify if current values deviate from threshold values specified in an alert definition. An email group can notify multiple users or target specific users during the time periods you specify.

Use the **Email Groups** tab to configure groups of SMTP email recipients. To open the **Email Groups** tab, in the PEM client, select **Management > Manage Alerts**. When the **Manage Alerts** tab opens, select **Email Groups** from the Quick Links toolbar.

The **Email Groups** tab displays a list of the currently defined email groups. To modify an existing group, select a group name and select **Edit** at the far left end of the row.

To define a new email group, select the plus sign (+) in the upper-right corner of the **Email Groups** table. Use the Email Groups dialog box to define an email group and its members.

Each row in the email group definition associates a set of email addresses with a specific time period. When an alert is triggered, the server evaluates the times specified in each row and sends the message to those group members whose definitions are associated with the time that the alert triggered.

1. Provide a name for the email group in the **Group Name** field.
2. To open the **Options** tab, select the plus sign (+) in the group members table.

3. Add the member addresses to receive notifications for the time period specified:

- Enter a comma-delimited list of recipient addresses in the **Reply to Addresses** field.
- Enter a comma-delimited list of addresses to receive a copy of the email in the **CC Addresses** field.
- Enter a comma-delimited list of addresses to receive a copy of the email without the knowledge of other recipients in the **Bcc Addresses** field.
- Enter the email address to send the messages from in the **From Address** field.
- Use the **Subject prefix** field to provide a message to add to the start of each subject line when a notification is sent.
- Use the **From Time** and **To Time** selectors to specify the time range for notifications to the group members that are identified on this row. Provide these values in the locale of the PEM client host. The PEM server translates the time into other time zones as required.

4. Select **Add** to add a row to the table, and specify another time period and the email addresses to notify during those hours.

5. When you've finished defining the email group, select **Save**.

After creating the email group, you can use the **Manage Alerts** tab to set up the notification details for an alert to direct notifications to the group.

### Deleting an email group

To delete an email group, in the Email Group table, select the name of the group and select **Delete**, located to the left of the group name.

The group name appears in the Email Group table in red. Select **Save** to permanently remove the group from the table.

## Webhook

You must configure the PEM Server to use webhooks to receive notification of alert events on threshold value violations in your configured applications.

### Creating a webhook

PEM monitors your system for conditions that require user attention. You can use a webhook to create the endpoints to receive a notification if current values deviate from threshold values specified in an alert definition. Based on the events triggered, PEM sends a notification to multiple webhook endpoints or to specific target webhook endpoints.

Use the **Webhooks** tab to configure endpoint recipients. To open the **Webhooks** tab, select **Management > Manage Alerts**. From the **Manage Alerts** tab, on the Quick Links toolbar, select **Webhooks**.

The **Webhooks** tab displays a list of the currently defined recipient applications as endpoints. Select an endpoint and select **Edit** at the far left end of the row to modify an existing endpoint.

To define a new webhook, select the plus sign (+) in the upper-right corner of the table. You can then use the **General** tab to define the basic details of the webhook:

- Provide a name for the webhook in the **Name** field.
- Specify a webhook URL to deliver all the notifications to in the **URL** field.
- Set the request method type used to make the call in the **Request Method** field: **POST** or **PUT**.
- By default, webhooks are enabled. To disable a webhook, set **Enable?** to **No**.

### Note

The **Enable?** setting works only if the `enable_webhook` parameter is set to `true` in the `agent.cfg` file. By default, the `enable_webhook` parameter is set to `true` only for the agent running on the PEM server host. For all other agents running on other hosts, you need to set it to `true` manually.



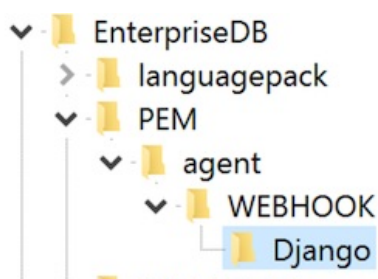
## Defining webhook SSL configurations

You can define the webhook SSL parameters in the respective agent configuration file or registry in Windows. You can find the list of webhook SSL parameters in [PEM agent configuration parameters](#). If you add or remove any of the agent configuration parameters, you must restart the agent to apply them.

- On Windows systems, PEM registry entries for webhooks are located in `HKEY_LOCAL_MACHINE\Software\EnterpriseDB\PEM\agent\WEBHOOK`.
- On Linux systems, PEM configuration options for webhooks are stored in the `agent.cfg` file, located by default in `/usr/edb/pem/agent/etc`.

### [WEBHOOK/Django]

```
webhook_ssl_key=<webhook_client_ssl_key_path>
webhook_ssl_cert=<webhook_client_ssl_certificate_path>
webhook_ssl_ca_cert=<webhook_server_ca_certificate_path>
webhook_ssl_crl=<crl_file_path_to_validate_webhook_server>
allow_insecure_webhooks=<true|false>
```



| Name                  | Type   | Data            |
|-----------------------|--------|-----------------|
| (Default)             | REG_SZ | (value not set) |
| AllowInsecureWebhooks | REG_SZ | true            |
| WebhookSSLCaCert      | REG_SZ | server_ca.pem   |
| WebhookSSLCert        | REG_SZ | client.crt      |
| WebhookSSLKey         | REG_SZ | client.key      |

Use the **HTTP Headers** tab to define the header parameters to pass while calling the webhook endpoints:

- Specify all the values as a key and value pair.
- Specify a key parameter in the **Key** field and a value in the **Value** field.
- To add HTTP headers, select the plus sign (+) in the upper-right corner of the HTTP Headers table.
- To delete HTTP headers, select **Delete** to the left of **Key**. The header remains in the list but in strike-through font. Select **Save** to permanently delete the headers.
- To edit the HTTP headers, select **Edit** to the left of **Key**.

Use the **Payload** tab to define the JSON data to send to the endpoint when an alert is triggered:

- Type** specifies data to send in the format type, that is, JSON.
- Use **Template** to configure JSON data sent to endpoints. In the template, you can use placeholders for the following:
  - `%AlertID%` — The id of the triggered alert.
  - `%AlertName%` — The name of the triggered alert.
  - `%ObjectName%` — The name of the server or agent on which the alert was triggered.
  - `%ObjectType%` — The type on which the alert was generated.
  - `%ThresholdValue%` — The threshold value reached by the metric when the alert triggered.
  - `%CurrentValue%` — The current value of the metric that triggered the alert.
  - `%CurrentState%` — The current state of the alert.
  - `%OldState%` — The previous state of the alert.
  - `%AlertRaisedTime%` — The time that the alert was raised or the most recent time that the alert state changed.
  - `%AgentID%` — The id of the agent by which the alert was generated.
  - `%AgentName%` — The name of the agent by which the alert was generated.
  - `%ServerID%` — The id of the server on which the alert was generated.
  - `%ServerName%` — The name of the server on which the alert was generated.
  - `%ServerIP%` — The ip or address of the server on which the alert was generated.
  - `%ServerPort%` — The the port of the server on which the alert was generated.

- `%DatabaseName%` — The name of the database on which the alert was generated.
- `%SchemaName%` — The name of the schema on which the alert was generated.
- `%PackageName%` — The name of the package on which the alert was generated.
- `%DatabaseObjectName%` — The name of the database object, like table name or function name, on which the alert was generated.
- `%Parameters%` — The list of custom parameters used to generate the alert.
- `%AlertInfo%` — The detailed database object-level information of the alert.

- Select **Test Connection** to test notification delivery to the mentioned endpoint.

Use the **Notifications** tab to specify an alert level for webhook endpoints:

- Set **All alerts** to **Yes** to enable all alert levels to send notifications.
- To send a notification when a specific alert level is reached, set the slider next to an alert level to **Yes**. You must set **All alerts** to **No** to configure an individual alert level.

### Deleting a webhook

To mark a webhook for deletion, in the Webhooks table, select the webhook name and select **Delete** to the left of the name. The alert remains in the list but in strike-through font.

**Delete** is a toggle. You can undo the deletion by selecting **Delete** a second time. Select **Save** to permanently delete the webhook definition.

## SNMP

You must configure the PEM server to send the notifications to an SNMP trap/notification receiver before notifications can be sent. Set the SNMP ver to v1, v2, or v3 to identify the SNMP version.

### Example - Configure `SNMP V3` traps with `net-snmp` trap receiver

1. Set `snmp_security_engine_id` to `PEM_SNMP_AGENT` in plain text format in the Server Configuration dialog box.
2. Convert the plain text value to hexadecimal format to use it in `snmptrapd.conf` file. (You can have hexadecimal values of `snmp_security_engine_id` up to 32 octets length).

```
echo PEM_SNMP_AGENT | hexdump -v -e '/1 "%02X"'
50454D5F534E4D505F4147454E540A
```

3. Set the following parameters in the Server Configuration dialog box:

- `snmp_security_nam` to `pem_snmp_user`
- `snmp_authentication_protocol` to `MD5`
- `snmp_authentication_password` to `pem_auth_pass`
- `snmp_privacy_protocol` to `DES`
- `snmp_privacy_password` to `pem_priv_pass`

4. The `snmptrapd.conf` file has the following values:

```
createUser -e 0x50454D5F534E4D505F4147454E540A pem_snmp_user MD5 pem_auth_pass DES pem_priv_pass
authUser log pem_snmp_user
```

## Using PEM with Nagios

The PEM server can send a passive alert result to Nagios network-alerting software when a user-defined alert is triggered. To instruct the PEM server to notify Nagios of a triggered alert, you must:

- Enable Nagios notification for each alert that triggers a notification from the PEM server to Nagios. You must configure PEM alerting before you create the `host.cfg` and `services.cfg` files.
- Configure Nagios-related behaviors of the PEM server.
- Create the `host.cfg` and `services.cfg` configuration files.
- If necessary, modify the Nagios configuration file and restart the Nagios server.
- Install the PEM agent on the system where the Nagios server is installed and register it with the PEM Server. Set `enable_nagios` configuration to `true` in the `agent.cfg` file for that agent, and restart the agent service.

After configuring the server to enable Nagios alerting, any triggered alerts send a passive check result to the Nagios service. The syntax of a passive alert is:

```
<timestamp> PROCESS_SERVICE_CHECK_RESULT; <host_name> ; <service_name> ; <service_status> ;
```

Where:

`timestamp` is the date and time that the alert was triggered.

`host_name` is the name of the server or agent.

`service_name` is the name of the alert.

`service_status` is the numeric service status value:

- 0 if the service status is OK
- 1 if the service status is WARNING
- 2 if the service status is CRITICAL
- 3 if the service status is UNKNOWN

The PEM server uses the following rules to evaluate the service status:

- If the PEM alert level is CLEARED, the warning message reads OK.
- If the PEM alert level is LOW, the warning message reads WARNING.
- If the `is_nagios_medium_alert_as_critical` flag (specified in the PEM server configuration dialog box) is set to FALSE and the alert level MEDIUM, the warning message reads WARNING.
- If the `is_nagios_medium_alert_as_critical` flag (specified in the PEM server configuration dialog box) is set to TRUE and the alert level is MEDIUM, the warning message reads CRITICAL.
- If the PEM alert level is HIGH, the warning message reads CRITICAL.

### Enabling Nagios notification for an alert

The PEM server maintains a unique set of notification properties for each enabled alert. Use the **Notification** tab of the **Manage Alerts** tab to specify that, when triggered, a given alert sends an alert notice to Nagios.

To modify the notification properties of an alert, right-click the name of the object monitored by the alert, and select **Management > Manage Alerts**. On the **Manage Alerts** tab, select **Edit** to the left of the alert name in the Alerts list. When the edit pane opens, select the **Notification** tab.

To enable Nagios notification, move the slider next to **Submit passive service check result to Nagios** to **Yes**. Then select **Save**.

### Configuring Nagios-related behavior of the PEM Server

You can use the Server Configuration dialog box to provide information about your Nagios configuration to the PEM server. To open dialog box, select **Management > Server Configuration**.

Four server configuration parameters specify information about your Nagios installation and PEM server behavior related to Nagios:

- Use the `nagios_cmd_file_name` parameter to specify the location of the Nagios pipeline file that receives passive check alerts from PEM. The default value of this parameter is `/usr/local/nagios/var/rw/nagios.cmd`. If your `nagios.cmd` file resides elsewhere, specify the file location in the **Value** field.
- Move the slider in the `nagios_enabled` parameter to **Yes** to instruct the PEM server to send passive check alerts to Nagios.
- Use the `nagios_medium_alert_as_critical` slider to specify the warning severity that the PEM server passes to Nagios if a medium alert is triggered:
  - If the `is_nagios_medium_alert_as_critical` flag is set to **FALSE** and the alert level is **MEDIUM**, the warning message reads **WARNING**.
  - If the `is_nagios_medium_alert_as_critical` flag is set to **TRUE** and the alert level is **MEDIUM**, the warning message reads **CRITICAL**.
- Use the `nagios_spool_retention_time` parameter to specify the number of days of notification history to store on the PEM server. The default value is 7 days.

After modifying parameter values, select **Save** in the upper-right corner of the dialog box.

### Creating the hosts.cfg and services.cfg file

The `templates.cfg` file (by default, located in `/usr/local/nagios/etc/objects`) specifies the properties of a generic host and generic service. The properties specify the parameters used in the `hosts.cfg` and `services.cfg` files.

In most cases (when PEM is installed in a default configuration), you don't need to modify the `templates.cfg` file before creating the `hosts.cfg` and `services.cfg` files. If necessary, you can modify the `templates.cfg` file to specify alternative values for parameters or to create new templates.

Before modifying the Nagios configuration file, use the following command to create a `hosts.cfg` file that contains information about the PEM hosts that reside on the local system:

```
psql -U postgres -p 5433 -d pem -A -t -c "select pem.create_nagios_host_config('generic-host')" > /usr/local/nagios/etc/objects/hosts.cfg
```

Then, use the following command to create a `services.cfg` file that contains information about the PEM services that reside on the local system:

```
psql -U postgres -p 5433 -d pem -A -t -c "select pem.create_nagios_service_config('generic-service')" > /usr/local/nagios/etc/objects/services.cfg
```

If you want to use a `custom template.cfg` file entry, specify the entry name in place of `generic-host` or `generic-service` in these commands.

### Modifying the Nagios configuration file

After creating the `host.cfg` and `services.cfg` files, you must specify their location in the Nagios configuration file (by default, `/usr/local/nagios/etc/nagios.cfg`). Modify the configuration file, adding entries that specify the location of the files:

```
cfg_file=/usr/local/etc/objects/hosts.cfg
```

```
cfg_file=/usr/local/etc/objects/services.cfg
```

You can use the following command to confirm that Nagios is properly configured:

```
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

After confirming that Nagios is configured correctly, restart the Nagios service:

```
/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
```

## 22.4 Capacity Manager

PEM's Capacity Manager analyzes collected statistics (metrics) to generate a graph or table that displays the historical usage statistics of an object. It can also project the anticipated usage statistics for an object. You can configure Capacity Manager to collect and analyze metrics for a specific host, server, database, or database object.

You can tailor the content of the Capacity Manager report by choosing:

- Specific metrics to include in the report
- The time range over which the metrics were gathered
- A high or low threshold for the metrics analyzed

You can also specify a start and end date for the Capacity Manager report. If the end date of the report specifies a time in the future, Capacity Manager analyzes the historical usage of the selected object to extrapolate the projected object usage in the future.

To open Capacity Manager, in the PEM client, select **Management > Capacity Manager**. The Capacity Manager wizard opens, displaying a tree on the **Metrics** tab. On the **Metrics** tab, expand the tree to review the metrics for the node that you want to analyze. Select the metric to include it in your report.

Capacity Manager uses the aggregation method specified with the **Aggregation** list. Capacity Manager uses the aggregation method to evaluate and plot the metric values. Select from:

- **Average** — Use the average of the values recorded during the time period.
- **Maximum** — Use the maximum value recorded during the time period.
- **Minimum** — Use the minimum value recorded during the time period.
- **First** — Use the first value recorded during the time period.

To remove a metric from the Capacity Manager report, clear the box to the left of the name of a metric.

Move the slider next to **Graph/chart metrics individually?** to **Yes** to produce a separate report for each metric selected on the **Metrics** tab. If the option is set to **No**, all selected metrics are merged into a single graph or table.

Select **Generate** to display the report onscreen (accepting the default configuration options), or use the **Options** tab to customize sampling boundaries, report type, and report destination. The times displayed on the **Options** tab are from the time zone where the PEM client resides.

Use the fields in the **Time Period** box to define the boundaries of the Capacity Manager report:

- Use the **Period** list to select the type of time period to use for the report. You can select:

|                          |  |
|--------------------------|--|
| Start time and end time  | Specify a start date and an end date/time for the report   |
| Start time and threshold | Specify a start date and time and a threshold to determine the end time and date for the report. |

| Start time and end time               | Specify a start date and an end date/time for the report  |
|---------------------------------------|---|
| Historical days and extrapolated days | Specify a start date for the report that is a number of days in the past and an end date that is a number of days in the future. This option is useful for report templates that don't specify fixed dates. |
| Historical days and threshold         | Specify a start date that is a number of days in the past, and end it when a threshold value is reached.  |

After specifying the type of time period for the report, select from other options in the **Time Period** box to define the time period for the report:

- Use the date and time selectors next to the **Start time** field to specify the starting date and time of the sampling period. Or select the number of historical days of data to include in the report. The date and time specified in the **Start time** field can't be later than the current date/time.

By default, Capacity Manager selects a start time one week prior to the current date and time.

- The end boundary for the report can be a time, a number of days in the future, or the point at which a selected metric reaches a specified threshold value. Use the date and time selectors next to **End time** to specify an end boundary for the report. Or select the number of extrapolated days of data to include in the report. The time specified in the **End time** field must be later than the time specified in the **Start time** field.

If you select an end date and time in the future, Capacity Manager uses historical usage information to extrapolate anticipated future usage. Since the projected usage is based on the sampling of historical data, the accuracy of the future usage trend improves with a longer sampling period.

To specify a threshold value, use the list in the **Threshold** field to select a metric, an operator (**Exceeds** or **Falls below**), and a target value for the metric. If you choose to define the end of the report using a threshold, the Capacity Manager report terminates when the value for the selected metric exceeds or falls below the specified value.

The `cm_max_end_date_in_years` configuration parameter defines a default time value for the end boundary of a Capacity Manager report. If you specify a threshold value as the end boundary of a report and the anticipated usage of the boundary isn't met before the maximum time passes, the report terminates at the time specified by the `cm_max_date_in_years` parameter. By default, `cm_max_end_date_in_years` is 5. You can use the Server Configuration dialog box to modify the value of `cm_max_end_date_in_years`.

The fields in the **Report** box specify the report type and destination. Use the **Include on report** options to specify the type of report produced by Capacity Manager:

- Select **Graph** to display the report in the form of a line graph in the PEM client window.
- Select **Table of data** to display a table containing the report data in the PEM client window.
- Select **Graph and table of data** to display both a line graph and a data table in the PEM client window.

Use the **Report destination** options to set where to display or save the report:

- Select **New tab** to display the report on a new tab in the PEM client. You must select **New tab** to display the first generation of a Capacity Manager report. For subsequent reports, you can select **Previous tab**.
- Select **Previous tab** to reuse a previously opened tab when displaying the report.
- Select **Download the report as a file** and specify a file name to write the report to the specified file.

After you specify the report boundaries and select the type and destination of the Capacity Manager report, select **Generate** to create the report.

Reports saved to file are stored as HTML. You can review a Capacity Manager report with any web browser that supports scalable vector graphics (SVG). Browsers that don't support SVG can't display a Capacity Manager graph and might include unwanted characters.

## Capacity Manager templates

After defining a report, you can save the definition as a template for future reports. All PEM users can access Capacity Manager report templates. To save a report definition as a template:

1. Use the **Metrics** and **Options** tabs to define your report.

- 2. Select **Save**.
- 3. In the **Save Template** dialog box, Provide a report name in the **Title** field and select a location to store the template in the tree.
- 4. Select **OK**.

When creating a report, you can use **Load Template** to browse and open an existing template. Once opened, you can modify the report definition and save it again, either as a new template or by overwriting the original template.

Select **Manage Templates** to open a dialog box that lets rename or remove unwanted templates.

Capacity Manager metrics - reference

The Capacity Manager metrics available vary by platform and are subject to change. The available metrics can include the metrics described in the table.

| Metric name                          | Description   |
|--------------------------------------|---|
| # Dead Tuples                        | The number of dead tuples in the selected table.  |
| # Dead Tuples+                       | The cumulative number of dead tuples in the selected table.   |
| # Heap Tuples Fetched by Index Scans | The number of heap tuples fetched by index scans.   |
| # Heap Tuples Fetched by Index Scans | The cumulative number of heap tuples fetched by index scans.  |
| # Idle Backends+                     | The cumulative number of currently idle backend clients.  |
| # Index Scans                        | The number of index scans performed on the specified object.  |
| # Index Scans+                       | The cumulative number of index scans performed on the specified object.   |
| # Index Tuples Read                  | The number of index tuples read.  |
| # Index Tuples Read+                 | The cumulative number of index tuples read.   |
| # Live Tuples                        | The number of tuples visible to transactions.   |
| # Live Tuples+                       | The cumulative number of tuples visible to transactions.  |
| # Pages Estimated by ANALYZE         | The number of pages estimated by ANALYZE.   |
| # Pages Estimated by ANALYZE+        | The cumulative number of pages estimated by ANALYZE.  |
| # Sequential Scans                   | The number of sequential scans performed on the specific table.   |
| # Sequential Scans+                  | The cumulative number of sequential scans performed on the specific table.  |
| # Sequential Scan Tuples             | The number of tuples sequentially scanned in the specific table.  |
| # Sequential Scan Tuples+            | The cumulative number of tuples sequentially scanned in the specific table.   |
| # Tuples Deleted                     | The number of tuples deleted.   |
| # Tuples Deleted+                    | The cumulative number of tuples deleted.  |
| # Tuples Estimated by ANALYZE        | The number of live (visible) tuples estimated by ANALYZE.   |
| # Tuples Estimated by ANALYZE+       | The cumulative number of live tuples estimated by ANALYZE.  |
| # Tuples HOT Updated                 | The number of tuples HOT updated. In a HOT update, the new tuple resides in the same block as the original tuple and the tuples share an index entry. |
| # Tuples HOT Updated+                | The cumulative number of tuples HOT updated.  |
| # Tuples Inserted                    | The number of tuples inserted into the specified table.   |
| # Tuples Inserted+                   | The cumulative number of tuples inserted into the specified table.  |
| # Tuples Updated                     | The number of tuples updated in the selected table.   |
| # Tuples Updated+                    | The cumulative number of tuples updated in the selected table.  |
| Blocks Hit                           | The number of blocks found in the cache.  |

| Metric name                      | Description   |
|----------------------------------|---|
| Blocks Hit+                      | The cumulative number of blocks found in the cache.   |
| Blocks Read                      | The number of blocks read.  |
| Blocks Read+                     | The cumulative number of blocks read.   |
| Blocks Read from InfiniteCache   | The number of blocks read from InfiniteCache.   |
| Blocks Read from InfiniteCache+  | The cumulative number of blocks read from InfiniteCache.  |
| Blocks Written                   | The number of blocks written.   |
| Blocks Written+                  | The cumulative number of blocks written.  |
| Buffers Allocated                | The number of buffers allocated.  |
| Buffers Allocated+               | The cumulative number of buffers allocated.   |
| Buffers Written - Backends       | The number of buffer blocks written to disk by server processe (processes connected to a client application). |
| Buffers Written - Backends+      | The cumulative number of buffer blocks written to disk by server processes.                                   |
| Buffers Written - Checkpoint     | The number of blocks written to disk by the checkpoint process.   |
| Buffers Written - Checkpoint+    | The cumulative number of blocks written to disk by the checkpoint process.                                    |
| Buffers Written - Cleaning Scan  | The number of blocks written to disk by the autovacuum process.   |
| Buffers Written - Cleaning Scan+ | The cumulative number of blocks written to disk by the autovacuum process.                                    |
| Bytes Received (KB)              | The number of bytes received from the client (in kilobytes).  |
| Bytes Received (KB)+             | The cumulative number of bytes received (in kilobytes).   |
| Bytes Sent (KB)                  | The number of bytes sent to the client (in kilobytes).  |
| Bytes Sent (KB)+                 | The cumulative number of bytes sent (in kilobytes).   |
| Checkpoints - Timed              | The number of checkpoint operations triggered by the checkpoint interval.                                     |
| Checkpoints - Timed+             | The cumulative number of checkpoint operations triggered by the checkpoint interval.                          |
| Checkpoints - Untimed            | The number of checkpoint operations triggered by checkpoint size.   |
| Checkpoints - Untimed+           | The cumulative number of checkpoint operations triggered by checkpoint size.                                  |
| Database Size (MB)               | The size of the specified database (in megabytes).  |
| Free RAM Memory                  | The amount of free RAM memory (in megabytes).   |
| Free Swap Memory                 | The amount of free swap space on disk (in megabytes).   |
| Heap Blocks Hit                  | The number of heap blocks found in the cache.   |
| Heap Blocks Hit+                 | The cumulative number of heap blocks found in the cache.  |
| Heap Blocks Read                 | The number of heap blocks read.   |
| Heap Blocks Read+                | The cumulative number of heap blocks read.  |
| Index Blocks Hit                 | The number of index blocks found in the cache.  |
| Index Blocks Hit+                | The cumulative number of index blocks found in the cache.   |
| Index Blocks Read                | The number of index blocks read.  |
| Index Blocks Read+               | The cumulative number of index blocks read.   |
| Index Size (MB)                  | The size of the specified index (in megabytes).   |
| In Packets Discards              | The number of inbound packets discarded.  |
| In Packets Discards+             | The cumulative number of inbound packets discarded.   |
| In Packets Errors                | The number of inbound packets that contain errors.  |
| In Packets Errors+               | The cumulative number of inbound packets that contain errors.   |
| Link Bandwidth (Mbit/s)          | The speed of the network adapter (in megabits per second).  |



| Metric name                           | Description  |
|---------------------------------------|--|
| Load Average - 15 Minute              | CPU saturation (in percent) - 15 minute sampling average.  |
| Load Average - 1 Minute               | CPU saturation (in percent) - 1 minute sampling average.   |
| Load Average - 5 Minute               | CPU saturation (in percent) - 5 minute sampling average.   |
| Load Percentage                       | CPU saturation in percent.   |
| Number of Prepared Transactions+      | The cumulative number of prepared transactions.  |
| Number of WAL Files+                  | The cumulative number of write-ahead log files.  |
| Out Packets Discards                  | The number of outbound packets discarded.  |
| Out Packets Discards+                 | The cumulative number of outbound packets discarded.   |
| Out Packets Errors                    | The number of outbound packets that contain errors.  |
| Out Packets Errors+                   | The cumulative number of outbound packets that contain errors.   |
| Packets Received                      | The number of packets received.  |
| Packets Received+                     | The cumulative number of packets received.   |
| Packets Sent                          | The number of packets sent.  |
| Packets Sent+                         | The cumulative number of packets sent.   |
| Size (MB)                             | The total size of the disk (in megabytes).   |
| Size of Indexes (MB)                  | The size of indexes on the specified table (in megabytes).   |
| Space Available (MB)                  | The current disk space available (in megabytes).   |
| Space Used (MB)                       | The current disk space used (in megabytes).  |
| Table Size (MB)                       | The size of the specified table (in megabytes).  |
| Tablespace Size (MB)                  | The size of the specified tablespace (in megabytes).   |
| Temp Buffers (MB)                     | The size of temporary buffers (in megabytes).  |
| Toast Blocks Hit                      | The number of TOAST blocks found in the cache.   |
| Toast Blocks Hit+                     | The cumulative number of TOAST blocks found in the cache.  |
| Toast Blocks Read                     | The number of TOAST blocks read.   |
| Toast Blocks Read+                    | The cumulative number of TOAST blocks read.  |
| Total RAM Memory                      | The total amount of RAM memory on the system (in megabytes).   |
| Total Swap Memory                     | The total amount of swap space on the system (in megabytes).   |
| Total Table Size w/Indexess and Toast | The total size of the specified table (including indexes and associated oversized attributes).   |
| Transactions Aborted                  | The number of aborted transactions.  |
| Transactions Aborted+                 | The cumulative number of aborted transactions.   |
| Transactions Committed                | The number of committed transactions.  |
| Transactions Committed+               | The cumulative number of committed transactions.   |
| Tuples Deleted                        | The number of tuples deleted from the specified table.   |
| Tuples Deleted+                       | The cumulative number of tuples deleted from the specified table.  |
| Tuples Estimated by ANALYZE           | The number of visible tuples in the specified table.   |
| Tuples Estimated by ANALYZE+          | The cumulative number of visible tuples in the specified table.  |
| Tuples Fetched                        | The number of tuples fetched from the specified table.   |
| Tuples Fetched+                       | The cumulative number of tuples fetched from the specified table.  |
| Tuples HOT Updated                    | The number of tuples HOT updated. In a HOT update, the new tuple resides in the same block as the original tuple and the tuples share an index entry.            |
| Tuples HOT Updated+                   | The cumulative number of tuples HOT updated. In a HOT update, the new tuple resides in the same block as the original tuple and the tuples share an index entry. |
| Tuples Inserted                       | The number of tuples inserted into the specified table.  |

| Metric name           | Description  |
|-----------------------|--|
| Tuples Inserted+      | The cumulative number of tuples inserted into the specified table. |
| Tuples Returned       | The number of tuples returned in result sets.                      |
| Tuples Returned+      | The cumulative number of tuples returned in result sets.           |
| Tuples Updated        | The number of tuples updated in the specified table.               |
| Tuples Updated+       | The cumulative number of tuples updated in the specified table.    |
| WAL Segment Size (MB) | The segment size of the write-ahead log (in megabytes).            |

#### Note

The + following the name of a metric means that the data for the metric is gathered cumulatively. Metrics that aren't followed by the '+' sign are collected as a point-in-time value.

## 22.5 Audit Manager

You can use the PEM Audit Manager to simplify audit log configuration for EDB Postgres Advanced Server instances. With Audit Manager, you can configure logging attributes such as:

- How often PEM collects log files
- The type of database activities that are included in the log files
- How often to rotate log files and when

Audit logs can include the following activities:

- All connections made to the database instance
- Failed connection attempts
- Disconnections from the database instance
- All queries (SELECT statements)
- All DML statements (INSERT, UPDATE, DELETE)
- All DDL statements (for example, CREATE, DROP, ALTER)

Once the audit logs are stored on the PEM server, you can use the Audit Log dashboard to review the information in an easy-to-read form. The Audit Log dashboard allows you to filter the log file by:

- Timestamp range (when an activity occurred)
- The database on which the activity occurred
- The user performing the activity
- The type of command being invoked

### Setting the EDB Postgres Advanced Server instance service ID

To configure logging for an EDB Postgres Advanced Server instance, the server must be a PEM-managed server with a bound agent, and the server registration must include the name of a service script. When registering a new server, include the service name in the **Service ID** field on the **Advanced** tab of the New Server dialog box.

Before adding a service name to an existing, registered and connected server, you must disconnect the server:

1. Right-click the server name, and select **Disconnect server** from the context menu.
2. Right-click the server name and select **Properties** from the context menu.
3. Select the **Advanced** tab, and add a service name to the **Service ID** field.

The Service ID field allows the PEM server to stop and start the service.

- The name of the Advanced Server 12 service script is `edb-as-12` .
- The name of the Advanced Server 11 service script is `edb-as-11` .
- The name of the Advanced Server 10 service script is `edb-as-10` .

## Setting the EDB Audit Configuration probe

Before configuring audit logging of EDB Postgres Advanced Server servers, you must ensure that the EDB Audit Configuration probe is enabled. To open the **Manage Probes** tab and check the status of the probe, right-click the name of a registered EDB Postgres Advanced Server server in the tree, and select **Management > Manage Probes**.

Ensure that the Enabled column in the Probe Configuration dialog box is set to **Yes** for the EDB Audit Configuration probe. If EDB Audit Configuration isn't enabled, use the **Enabled?** switch on the **Manage Probes** tab to enable it.

## Configuring audit logging with the Audit Manager

1. To open the Audit Manager wizard, select **Management > Audit Manager**. The Audit Manager - Welcome dialog box opens. Select **Next** to continue.
2. Use the Select Servers tree to specify the servers to which to apply the auditing configuration. To make a server available in the tree, you must provide the Service ID on the **Advanced** tab of the Create Server dialog box when registering a server for monitoring by PEM. Only EDB Postgres Advanced Server supports auditing. PostgreSQL servers don't appear in the tree.

Select **Next** to continue.

3. The Auditing Parameters Configuration dialog box lets you enable or disable auditing and choose how often log records are collected into PEM. Use the Auditing Parameters Configuration dialog box to specify auditing preferences:
  - Use the **Auditing** switch to enable or disable auditing on the specified servers.
  - Use the **Audit destination** list to select a destination for the audit logs. Select **File** or **Syslog**. This feature is supported only on EDB Postgres Advanced Server 10 and later releases.
  - Use the **Import logs to PEM** switch to periodically import log records from each server to the PEM Server. Set the switch to **Yes** to import log files. The default is **No**.
  - Use the **Collection frequency** list to specify how often PEM collects log records from monitored servers when log collection is enabled.
  - Use the **Log format** list to select the raw log format to write on each server. If log collection is enabled, the PEM server uses CSV format.
  - Use the **File name** field to specify the format used when generating log file names. By default, the format is set to `audit-%Y-%m-%d_%H%M%S` where:
    - `audit` is the file name specified in the Audit Directory Name field
    - `Y` is the year that the log was stored
    - `m` is the month that the log was stored
    - `d` is the day that the log was stored
    - `H` is the hour that the log was stored
    - `M` is the minute that the log was stored
    - `S` is the second that the log was stored
  - Select **Change Log Directory for selected servers?** and use the **Audit Directory Name** field to specify a directory name to which to write the audit logs. The directory resides beneath the data directory on the PEM server.
  - Use the **Log directory** box to specify information about the directory in which the log files are saved:

- Move the **Change log directory for selected servers?** switch to **Yes** to enable the **Directory name** field.
- Use the **Directory name** field to specify the name of the directory on each server to write audit logs to. The directory specified is created as a subdirectory of the data directory on the server.

Select **Next** to continue.

4. The Audit log configuration dialog box is available only if you enabled auditing on the Auditing Parameters Configuration dialog box. Use the Audit Log Configuration dialog box to specify log configuration details to apply to each server:

- Use the **Connection attempts** switch to specify whether to log connection attempts. Specify:

**None** to disable connection logging

**All** to indicate that all connection attempts are logged

**Failed** to log any connection attempts that fail

- Use the **Disconnection attempts** switch to specify whether to log disconnections. Specify:

**None** if you don't want to log disconnections

**All** to enable disconnection logging

- Use the **Log statements** field to specify the statement types that are logged. Select from:

**Select** — Log all statements that include the **SELECT** keyword.

**Error** — Log all statements that result in an error.

**DML** — Log all DML (data modification language) statements.

**DDL** — Log all DDL (data definition language) statements, that is, those that add, delete or alter data.

Select **Select All** to select all statement types.

Select **Unselect All** to clear all statement types.

- Use the **Audit tag** field to specify a tracking tag for the collected logs. Audit tagging functionality is available only for supported versions of EDB Postgres Advanced Server.

- Use the **Log rotation** box to specify how to manage the log files on each server:

Use the **Enable?** switch to rotate log files. Use a new log file periodically to prevent a single file from becoming too large.

Use the **Day** list to select days on which to rotate the log file.

Use the **Size (MB)** field to specify a size in megabytes at which to rotate the log file.

Use the **Time (seconds)** field to specify the number of seconds between log file rotations.

Select **Next** to continue:

5. Use the Schedule Auditing Changes dialog box to determine when auditing configuration changes take effect.

- Select **Configure logging now?** if you want the auditing configuration changes to take place immediately. The affected database servers

restart so the auditing changes can take effect.

- Use the **Time?** selector to schedule the auditing configuration changes to take place at some point in the future. Select the desired date and time from the lists. The affected database servers restart at the specified date and time to put the auditing changes into effect.

Select **Finish** to complete the auditing configuration process.

Audit Manager schedules a job to apply the configuration to each server. The job consists of two tasks: one to update the audit logging configuration on the server and one to restart the server with the new configuration.

You can use the **Scheduled Tasks** tab to review a list of scheduled jobs. To open the **Scheduled Tasks** tab, select the name of a server or agent and select **Management > Scheduled Tasks**.

## Viewing the log with the Audit Log dashboard

Use the Audit Log dashboard to view the audit log from EDB Postgres Advanced Server database instances.

To open the Audit Log dashboard, right-click a server or agent node and select **Audit Log Analysis** from the **Dashboards** menu. The Audit Log dashboard displays the audit records in reverse chronological order.

| Properties SQL Statistics Dependencies Dependents Monitoring <u>Audit Log</u>  |         |                      |              |               |            |               |                |                 |                |  |
|--|---------|----------------------|--------------|---------------|------------|---------------|----------------|-----------------|----------------|--|
| Postgres Enterprise Manager Host Audit Log   |         |                      |              |               |            |               |                |                 |                |  |
| Object Type Host Agent Status UP ( Since : 15/04/2020, 19:35:06 ) Generated On 29/04/2020, 10:08:32 No. of alerts None |         |                      |              |               |            |               |                |                 |                |  |
| Audit Logs   |         |                      |              |               |            |               |                |                 |                |  |
| id   | Server  | Timestamp            | User Name    | Database Name | Process ID | Session ID    | Transaction ID | Connection From | Command        | Message  |
| 7879   | EPAS_12 | 29/04/2020, 10:00:07 | enterprisedb | postgres      | 26391      | 5ea902cf.6717 | 0              | 127.0.0.1:46780 | idle           | disconnection: session time: 0:00:00.014 user=enterprisedb database=postgres host=127.0.0.1 port=46780 |
| 7878   | EPAS_12 | 29/04/2020, 10:00:07 | enterprisedb | postgres      | 26391      | 5ea902cf.6717 | 0              | 127.0.0.1:46780 | idle           | statement: SELECT setting FROM pg_settings WHERE name = 'edb_audit_rotation_seconds'                   |
| 7877   | EPAS_12 | 29/04/2020, 10:00:07 | enterprisedb | postgres      | 26391      | 5ea902cf.6717 | 0              | 127.0.0.1:46780 | idle           | statement: SELECT version();   |
| 7876   | EPAS_12 | 29/04/2020, 10:00:07 | enterprisedb | postgres      | 26391      | 5ea902cf.6717 | 0              | 127.0.0.1:46780 | authentication | connection authorized: user=enterprisedb database=postgres   |
| 7875   | EPAS_12 | 29/04/2020, 10:00:07 | enterprisedb | postgres      | 26388      | 5ea902cf.6714 | 0              | 127.0.0.1:46774 | idle           | disconnection: session time: 0:00:00.007 user=enterprisedb database=postgres host=127.0.0.1 port=46774 |
| 7874   | EPAS_12 | 29/04/2020, 10:00:07 | enterprisedb | postgres      | 26388      | 5ea902cf.6714 | 0              | 127.0.0.1:46774 | authentication | connection authorized: user=enterprisedb database=postgres   |
| 7873   | EPAS_12 | 29/04/2020, 10:00:07 | enterprisedb | postgres      | 26384      | 5ea902cf.6710 | 0              | 127.0.0.1:46766 | idle           | disconnection: session time: 0:00:00.029 user=enterprisedb database=postgres host=127.0.0.1 port=46766 |
| 7872   | EPAS_12 | 29/04/2020, 10:00:07 | enterprisedb | postgres      | 26384      | 5ea902cf.6710 | 0              | 127.0.0.1:46766 | idle           | statement: SELECT setting FROM pg_settings WHERE name='log_temp_files'                                 |
| 7871   | EPAS_12 | 29/04/2020, 10:00:07 | enterprisedb | postgres      | 26384      | 5ea902cf.6710 | 0              | 127.0.0.1:46766 | idle           | statement: SELECT setting FROM pg_settings WHERE name='log_autovacuum_min_duration'                    |
| 7870   | EPAS_12 | 29/04/2020, 10:00:07 | enterprisedb | postgres      | 26384      | 5ea902cf.6710 | 0              | 127.0.0.1:46766 | idle           | statement: SELECT setting FROM pg_settings WHERE name='log_min_duration_statement'                     |
| 7869   | EPAS_12 | 29/04/2020, 10:00:07 | enterprisedb | postgres      | 26384      | 5ea902cf.6710 | 0              | 127.0.0.1:46766 | idle           | statement: SELECT (setting::int/(24*60))::int FROM pg_settings WHERE name = 'log_rotation_age'         |
| 7868   | EPAS_12 | 29/04/2020, 10:00:07 | enterprisedb | postgres      | 26384      | 5ea902cf.6710 | 0              | 127.0.0.1:46766 | idle           | statement: SELECT (setting::int/1024)::int FROM pg_settings WHERE name = 'log_rotation_size'           |
| 7867   | EPAS_12 | 29/04/2020, 10:00:07 | enterprisedb | postgres      | 26384      | 5ea902cf.6710 | 0              | 127.0.0.1:46766 | idle           | statement: SELECT upper(setting) FROM pg_settings WHERE name='syslog_facility';                        |

You can use filtering to limit the number of audit records that are displayed. Select **Show Filters** to expose the filters panel. Use the filters panel to provide selection criteria for the audit records you want to display.

- Use the **Start** field to specify a start date for the report.
- Use the **End** field to specify an end date for the report.
- Use the **User** field to display only those entries where the activity was started by the given Postgres user.
- Use the **Database** field to display only those entries where the activity was issued on the given database.

- Use the **Command type** field to display only those entries where the activity was of the given type. Command types you can specify are **idle**, **authentication**, and **SELECT**. (For viewing SQL statements from user applications, specify the idle command type.)

Select **Filter** to apply the filtering criteria to the log entries.

## 22.6 Log Manager

You can use the PEM Log Manager to simplify server log configuration for Postgres instances. With Log Manager, you can modify all of your server log parameters including:

- Where log files are written
- How often log files are written
- The type of information written to log files
- The format of log file entries
- Log rotation properties

To configure logging for a Postgres instance, you must register the server as a PEM-managed server, and the registration information must include the name of a service script.

1. To open Log Manager, in the PEM client, select **Management > Log Manager**. The wizard opens and welcome message appears. Select **Next**.
2. The Server Selection dialog box displays a list of the server connections monitored by PEM. Select the check boxes next to the names of servers to which you want the Log Manager wizard to apply the specified configuration. Log Manager is disabled for any server displaying a red exclamation mark to the left of its name in the Server selection tree. A server might not be enabled for several reasons:
  - Log Manager can configure only a server that specifies a service ID on the **Advanced** tab of the Properties dialog box. To provide a service ID:
    1. In the tree, right-click the server name and select **Disconnect Server** from the context menu.
    2. If prompted, provide a password.
    3. From the context menu for the server, select **Properties**.
    4. On the **Advanced** tab, enter the name of the service in the **Service ID** field.
    5. Select **Save**.
  - If the PEM agent bound to the server doesn't have enough privileges to restart the server, the server is disabled.
  - If the PEM agent bound to the server is an older version than the associated PEM server, the server is disabled.

Select **Next**.

3. Use the Log configuration dialog box to specify how often to import log files to PEM and to specify log rotation details.

Options in the **Import Logs** box specify how often to import the log files to PEM:

- Use the switch next to **Import logs to PEM** to specify whether to import the log files to PEM and to display them on the Server Log Analysis dashboard.
- Use the **Import Frequency** list to specify how often to import log files to PEM.

Use the **Log rotation configuration** box to specify the maximum length (lifespan or size) of a log file:

- Use the **Rotation Size** field to specify the maximum size of a log file in megabytes. The default value is 10 MB. Set to 0 if you don't want to place a limit on the maximum size of a log file.
- Use the **Rotation Time** field to specify the number of whole days to store in each log file. The default value is 1 day.

Use the **Truncation on Rotation** switch to specify server behavior for time-based log file rotation:

- Select **On** to overwrite any existing log file that has the same name as a new file.
- Select **Off** to append any new log file entries to an existing log file with the same name as a new log file. This is the default behavior.

Select **Next**.

4. Use the Where to Log dialog box to specify where to write log files:

- Select an option from the **Log Destination** box to specify a destination for the server log output:
  - Set the **stderr** switch to **Yes** to write log files to stderr.
  - Set the **csvlog** switch to **Yes** to write log files in a comma-separated value format. This option is enabled and not editable if you select **Import logs to PEM** on the Schedule dialog box. If you aren't importing server log files to PEM, you can edit this option.
  - Set the **syslog** switch to **Yes** to write log files to the system log files.
  - On Windows, set the **eventlog** switch to **Yes** to write log files to the event log.
- Use the **Log collection** box to specify your collection preferences:
  - Set the **Log Collector** switch to **Enable** to redirect captured log messages (directed to stderr) into log files.
  - Set the **Log Silent Mode** switch to **Enable** to run the server silently in the background, disassociated from the controlling terminal.
- Use the **Log Directory** box to specify log file location preferences:
  - Set the **Change log directory for selected servers?** switch to **Yes** to maintain each set of log files in a separate directory.
  - Use the **Directory name** field to specify the directory to which to write log files. The directory resides beneath the **pg\_log** directory under the installation directory of the monitored server.
- Use the **Log File Name** field to specify a format for the log file name. If set to **Default**, the format is **enterprisedb-%Y-%m-%d\_%H%M%S**, where:
  - **enterprisedb** is the file name prefix
  - **Y** is the year that the log was stored
  - **m** is the month that the log was store
  - **d** is the day that the log was stored
  - **H** is the hour that the log was stored
  - **M** is the minute that the log was store
  - **S** is the second that the log was stored
- When logging to syslog is enabled:
  - Use the **Syslog Facility** list to specify the syslog facility to use.
  - Use the **Syslog Ident** field to specify the program name that identifies EDB Postgres Advanced Server entries in system logs.

Select **Next**.

5. Use the When to Log dialog box to specify the events that initiate a log file entry. The severity levels in order of severity, from most severe to least severe, are:

- **panic** — Errors that cause all database sessions to abort.
- **fatal** — Errors that cause a session to abort.
- **log** — Information messages of interest to administrators.
- **error** — Errors that cause a command to abort.
- **warning** — Error conditions in which a command completes but might not perform as expected.
- **notice** — Items of interest to users. This is the default.
- **info** — Information implicitly requested by the user.
- **debug5** through **debug1** — Detailed debugging information useful to developers.
- Use the **Client min messages** list to specify the lowest severity level of message sent to the client application.
- Use the **Log min messages** list to specify the lowest severity level to write to the server log.
- By default, when an error message is written to the server log, the text of the SQL statement that initiated the log entry isn't included. Use the **Log min error statement** list to specify a severity level that triggers SQL statement logging. If a message is of the specified severity or higher, the SQL statement that produced the message is written to the server log.
- Use the **Log min duration statement** list to specify a statement duration, in milliseconds. Any statements that exceed the specified number of milliseconds are written to the server log. A value of **-1** disables all duration-based logging. A value of **0** logs all statements and their duration.
- Use the **Log temp files** field to specify a file size, in kilobytes. When a temporary file reaches the specified size, it's logged. Specify a value of-

- 1 (the default) to disable this functionality.
- Use the **Log autoVacuum min duration** field to specify a time length, in milliseconds. If auto-vacuuming exceeds the length of time specified, the activity is logged. Specify a value of -1 (the default) to disable this functionality.

Select **Next**.

- Use the What to Log dialog box to specify log entry options that are useful for debugging and auditing.

The switches in the **Debug options** box instruct the server to include information in the log files related to query execution that might be of interest to a developer:

- Set the **Parse tree** switch to **Yes** to include the parse tree in the log file.
- Set the **Rewriter output** switch to **Yes** to include query rewriter output in the log file.
- Set the **Execution plan** switch to **Yes** to include the execution plan for each executed query in the log file.

When the **Indent Debug Options Output in Log** switch is set to **Yes**, the server indents each line that contains a parse tree entry, a query rewriter entry, or query execution plan entry. While indentation makes the resulting log file more readable, it results in a longer log file.

Use the switches in the **General Options** box to include auditing information in the log file:

- Set the **Checkpoints** switch to **Yes** to include checkpoints and restartpoints in the server log.
- Set the **Connections** switch to **Yes** to include each attempted connection to the server as well as successfully authenticated connections in the server log.
- Set the **Disconnections** switch to **Yes** to include a server log entry for each terminated session that provides the session information and session duration.
- Set the **Duration** switch to **Yes** to include the amount of time required to execute each logged statement in the server log.
- Set the **Hostname** switch to **Yes** to include both the IP address and host name in each server log entry. By default, only the IP address is logged. This might cause a performance penalty.
- Set the **Lock Waits** switch to **Yes** to write a log entry for any session that waits longer than the time specified in the `deadlock_timeout` parameter to acquire a lock. This is useful when trying to determine if lock waits are the cause of poor performance.

Use the **Error verbosity** list to specify the detail written to each entry in the server log:

- Select **default** to include the error message, DETAIL, HINT, QUERY, and CONTEXT in each server log entry.
- Select **terse** to log only the error message.
- Select **verbose** to include the error message, the DETAIL, HINT, QUERY, and CONTEXT error information, SQLSTATE error code and source code file name, the function name, and the line number that generated the error.

Use the **Prefix string** field to specify a printf-style string that is written at the beginning of each log file entry. For information about the options supported, see the `log_line_prefix` documentation in the [Postgres core documentation](#).

Use the **Statements** list to specify the SQL statements to include in the server log:

- Specify **none** (the default) to disable logging of SQL statements.
- Specify **ddl** to instruct to log ddl (data definition language) statements, such as CREATE, ALTER, and DROP.
- Specify **mod** to log all ddl statements as well as all dml (data modification language) statements, such as INSERT, UPDATE, DELETE, TRUNCATE, and COPY FROM.
- Specify **all** to log all SQL statements.

Select **Next**.

- Use the Schedule Logging Changes dialog box to specify when logging applies configuration changes:

- Set the **Configure logging now** switch to **Yes** to enable your configuration preferences. The server restarts when you complete the Log Manager wizard.
- Set **Configure logging now** to **No** to use the **Schedule it for some other time** calendar selector to specify a convenient time to apply logging configuration preferences and for the server to restart.

When you apply the configuration changes specified by the Log Manager wizard, the server restarts, temporarily interrupting use of the database server for users.



8. Select **Finish** to exit the wizard. Either restart the server or schedule the server restart for the time specified on the scheduling dialog box.

## Reviewing the Server Log Analysis dashboard

After invoking the Log Manager wizard and importing your log files to PEM, you can use the Server Log Analysis dashboard to review the log files for a selected server. To open the Server Log Analysis dashboard, right-click the name of a monitored server in the PEM client tree and select **Dashboards > Server Log Analysis**.

| Properties SQL Statistics Dependencies Dependents Monitoring <u>Server Log</u>   |                      |           |               |            |               |                |                 |         |  |
|--|----------------------|-----------|---------------|------------|---------------|----------------|-----------------|---------|--|
| Postgres Enterprise Manager Host > Postgres Enterprise Manager Server > Server Log   |                      |           |               |            |               |                |                 |         |  |
| Object Type Server Status UP ( Since : 27/04/2020, 15:47:09 ) Generated On 29/04/2020, 10:20:00 No. of alerts 6 ( Acknowledged : 0 ) |                      |           |               |            |               |                |                 |         |  |
| Server Logs  |                      |           |               |            |               |                |                 |         |  |
| id   | Timestamp            | User Name | Database Name | Process ID | Session ID    | Transaction ID | Connection From | Command | Message  |
| 1870601  | 29/04/2020, 10:15:06 | agent1    | pem           | 64373      | 5ea8fbc6.fb75 | 5002111        | 127.0.0.1:55512 | COPY    | duration: 0.187 ms statement: BEGIN;COPY pemdata.server_logs(server_id, log_time, user_name, database_name, process_id, connection_from, session_id, session_line_num, command_tag, session_start_time, virtual_transaction_id, transaction_id, error_severity, sql_state_code, message, detail, hint, internal_query, internal_query_pos, context, query, query_pos, location, application_name) FROM STDIN WITH NULL AS 'NULL' QUOTE "" CSV; |
| 1870600  | 29/04/2020, 10:15:06 | agent1    | pem           | 64373      | 5ea8fbc6.fb75 | 0              | 127.0.0.1:55512 | COMMIT  | duration: 0.572 ms statement: END;   |
| 1870599  | 29/04/2020, 10:15:06 | agent1    | pem           | 64373      | 5ea8fbc6.fb75 | 5002110        | 127.0.0.1:55512 | UPDATE  | duration: 0.129 ms statement: UPDATE pem.log_configuration SET (last_read_filename, file_offset) = ('/var/lib/pgsql/12/data/log/postgresql-2020-04-29_004252.csv', 3135795) WHERE server_id = 1;   |
| 1870598  | 29/04/2020, 10:15:06 | agent1    | pem           | 64373      | 5ea8fbc6.fb75 | 5002110        | 127.0.0.1:55512 | COPY    | duration: 0.289 ms statement: BEGIN;COPY pemdata.server_logs(server_id, log_time, user_name, database_name, process_id, connection_from, session_id, session_line_num, command_tag, session_start_time, virtual_transaction_id, transaction_id, error_severity, sql_state_code, message, detail, hint, internal_query, internal_query_pos, context, query, query_pos, location, application_name) FROM STDIN WITH NULL AS 'NULL' QUOTE "" CSV; |
| 1870597  | 29/04/2020, 10:15:06 | agent1    | pem           | 64373      | 5ea8fbc6.fb75 | 0              | 127.0.0.1:55512 | COMMIT  | duration: 0.521 ms statement: END;   |
| 1870596  | 29/04/2020, 10:15:06 | agent1    | pem           | 64373      | 5ea8fbc6.fb75 | 5002109        | 127.0.0.1:55512 | UPDATE  | duration: 0.132 ms statement: UPDATE pem.log_configuration SET (last_read_filename, file_offset) = ('/var/lib/pgsql/12/data/log/postgresql-2020-04-29_004252.csv', 3134305) WHERE server_id = 1;   |
| 1870595  | 29/04/2020, 10:15:06 | agent1    | pem           | 64373      | 5ea8fbc6.fb75 | 5002109        | 127.0.0.1:55512 | COPY    | duration: 0.200 ms statement: BEGIN;COPY pemdata.server_logs(server_id, log_time, user_name, database_name, process_id, connection_from, session_id, session_line_num, command_tag, session_start_time, virtual_transaction_id, transaction_id, error_severity, sql_state_code, message, detail, hint, internal_query, internal_query_pos, context, query, query_pos, location, application_name) FROM STDIN WITH NULL AS 'NULL' QUOTE "" CSV; |
| 1870594  | 29/04/2020, 10:15:06 | agent1    | pem           | 64373      | 5ea8fbc6.fb75 | 0              | 127.0.0.1:55512 | COMMIT  | duration: 0.382 ms statement: END;   |

The header information on the Server Log Analysis dashboard displays the date and time that the server was started, the date and time that the page was last updated, and the current number of triggered alerts.

Entries in the Server Log table appear in chronological order, with the most recent log entries first. Use the scroll bars to navigate through the log entries or to view columns that are off of the display.

Headings at the top of the server log table identify the information stored in each column. Hover over a column heading to view a tooltip that contains a description of the content of each column.

You can use filtering to limit the number of server log records that are displayed. Select **Show Filters** to open the filters panel and define a filter. Use the **filter definition** box to describe the selection criteria for selecting a subset of a report to display:

- Use the **From** field to specify a starting date for the displayed server log.
- Use the **To** field to specify an ending date for the displayed server log.
- Enter a role name in the **Username** field to display transactions performed only by that user.
- Enter a database name in the **Database** field to limit the displayed records to transactions that were performed against the specified database.
- Use the **Command Type** field to specify selection criteria for the commands that appear in the filtered report.

After you specify the criteria for filtering the server logs, select **Filter** to display the filtered server log in the Server Log table.

## Postgres Log Analysis Expert

The PEM Log Analysis Expert analyzes the log files of servers that are registered with Postgres Enterprise Manager and produces an analysis of your Postgres cluster's usage based on log file entries. You can use information on the Log Analysis Expert reports to make decisions about optimizing your cluster usage and configuration to improve performance.

Before using the PEM Log Analysis Expert, you must specify a **Service ID** value on the **Advanced** tab of the Server Properties dialog box. You must also use the Log Manager wizard to enable log collection by the PEM server.

1. To open the Postgres Log Analysis Expert wizard, in the PEM client, select **Management > Postgres Log Analysis Expert**. In the wizard's Welcome screen, select **Next**.

The wizard's Analyzer selection dialog box displays a list of analyzers from which you can select. Each analyzer generates a corresponding table, chart, or graph that contains information gleaned from the log files.

2. Select the check box next to an analyzer to specify for the Log Analysis Expert to prepare the corresponding table, chart, or graph. After making your selections, select **Next** to continue to the Server selection tree.

Use the tree to specify the servers you want the Postgres Log Analysis Expert to analyze. If you select multiple servers, the resulting report contains the corresponding result set for each server in a separate but continuous list. Select **Next** to continue to the Report options dialog box.

Use the **Options** section to specify the analysis method and the maximum length of any resulting tables:

- Use the **Aggregate method** list to select the method used by the Log Analysis Expert to consolidate data for the selected time span:
  - **SUM** calculates a value that is the sum of the collected values for the specified time span.
  - **AVG** calculates a value that is the average of the collected values for the specified time span.
  - **MAX** uses the maximum value that occurs in a specified time span.
  - **MIN** uses the minimum value that occurs in a specified time span.
- Use the **Time span** field to specify the number of minutes that the analyzer incorporates into each calculation for a point on a graph. For example, if the time span is five minutes, and the aggregate method is **AVG**, each point on the given graph contains the average value of the activity that occurred in a five-minute time span.
- Use the **Rows limit** field to specify the maximum number of rows to include in a table.

Use the **Time Intervals** section to specify the time range for the Log Analysis Expert to analyze:

- Set **Relative days** to **Yes** to enable the **(+/-) From date** field and specify the number of days before or after the date and time selected in the **From** field.
- Use the **From** field to specify the starting date and time for the analysis.
- Use the **To** field to specify the ending date and time for the analysis.
- Use the **(+/-) From date** selector to specify the number of days before or after the **From** date to include in the analysis.

After you specify the report options, select **Next** to continue to the Report destination dialog box.

3. You can select the default option and select **Finish** to view the Log Analysis Expert report in the PEM client's tabbed browser. Or select **Download the report** to save a copy of the report to an HTML file for later use.

## Reviewing the Postgres Log Analysis Expert report

If you choose to review the report immediately, the Postgres Log Analysis Expert report is displayed in the PEM Client window. The report header displays:

- The date and time that the report was generated

- The time period that the report spans
- The aggregation method specified when defining the report.

The name of the server for which information is displayed appears at the start of each section of the report.

The report displays the tables, graphs, and charts selected in the Log Analysis Expert wizard. Select **Jump To**, located in the lower-right corner of the screen, to navigate to a specific graphic.

Postgres Log Analysis Expert

Interval: 2020-4-20 15:55:19 - 2020-4-27 15:55:19    Generated: 2020-04-27 16:00:51    Span: 5 Minutes    Aggregate: SUM    Go to: Postgres Enterprise Manager Server

Postgres Enterprise Manager Server(192.168.1.18:5432)

Summary Statistics

| Settings                    | Values                               |
|-----------------------------|--------------------------------------|
| Number of unique queries    | 151649                               |
| Total queries               | 155045                               |
| Total queries duration      |                                      |
| First query                 | 27/04/2020 15:47:09.86 IST           |
| Last query                  | 27/04/2020 15:53:59.611 IST          |
| Queries peak time           | 27/04/2020 15:49:38 IST queries 2821 |
| Number of events            | 155045                               |
| Number of unique events     | 1                                    |
| Total number of sessions    | 348                                  |
| Total duration of sessions  |                                      |
| Average sessions duration   |                                      |
| Total number of connections | 0                                    |
| Total number of databases   | 0                                    |

Hourly DML Statistics

| Time             | Database name | Statement | Count | Min duration | Max duration | Avg duration |
|------------------|---------------|-----------|-------|--------------|--------------|--------------|
| 27/04/2020 15:00 | db01          | SELECT    | 80    | 0.05         | 105.40       | 4.90         |
| 27/04/2020 15:00 | edbstore_temp | SELECT    | 58    | 0.02         | 66.58        | 4.64         |
| 27/04/2020 15:00 | hr            | SELECT    | 48    | 0.01         | 29.26        | 1.87         |
| 27/04/2020 15:00 | pem           | COPY      | 1641  | 0.65         | 43.81        | 2.08         |
| 27/04/2020 15:00 | pem           | DELETE    | 73    | 0.44         | 8.74         | 1.04         |
| 27/04/2020 15:00 | pem           | INSERT    | 190   | 0.06         | 9.50         | 2.19         |

If the report contains an analysis of more than one monitored server, charts and tables are displayed in sets. First the graphs, tables, and charts that display statistics for one server appear. Then the graphics for the next server in the report appear.

22.7 Charts

You can use the **Manage Charts** tab to create or modify a custom line chart or table or to import a Capacity Manager template to use in a custom chart. After defining a chart, you can display the chart on a custom dashboard.

To open the **Manage Charts** tab, in the PEM client, select **Management > Manage Charts**. The **Manage Charts** tab provides a Quick Links menu for accessing dialog boxes to:

- Create a new chart for use on a custom dashboard.
- Import a Capacity Manager template to use as a template for creating a custom chart.
- Access online help.

The Custom Charts table displays a list of charts you defined. When a chart is new, the font is green. When you add a chart or refresh the screen, the font is black. Use the search box in the upper-right of the Custom Charts table to search through your custom charts. Specify:

- Chart name
- Type
- Level
- Metrics category

Use icons to the left of a chart name in the Custom Charts table to manage a chart:

- Select **Edit** to open the Chart Configuration wizard and modify aspects of the chart or table.
- Select **Delete** to delete the selected chart.

## Creating a custom chart

Select **Create New Chart** in the Quick Links section of the **Manage Charts** tab to open the Create Chart wizard. The wizard takes you through the steps required to define a new chart.

1. Use the Configure Chart dialog box to specify general information about the chart:
  - Specify the name of the chart in the **Name** field.
  - Use the list in the **Category** field to specify the category in which to display this chart. When adding a custom chart to a custom dashboard, you can select the chart from the category specified.
  - Use the **Type** field to specify if the chart is a line chart or a table.
  - Provide a description of the chart in the **Description** field. The description is displayed on a custom dashboard when the user selects the information icon next to the chart.

After you complete the fields on the Configure Chart dialog box, select **Next**.

2. Use the Select Metrics dialog box to select the metrics to display on the chart.
  - Use the **Metric level** list to specify the level of the PEM hierarchy from which you want to select metrics. You can specify **Agent**, **Database**, or **Server**. Each level offers access to a unique set of probes and metrics.
  - Use the tree in the **Available metrics** box to select the metrics to display on the chart.
 

If you're creating a table, you can select metrics from only one probe. Each node of the tree lists the metrics returned by a single probe. Expand a node of the tree and select the check box next to a metric name to include that metric data in the table.

If you're creating a line chart, expand the nodes of the tree and double-click each metric that you want to include in the chart.
  - Use the Selected Metrics panel to specify how the metric data appears in your chart. The selection panel displays the name of the metric in the nonmodifiable Metric [Probe] column. You can:
    - Select the trash can to delete a metric from the list of selected metrics.
    - Use the lists in the Selection Criteria column to specify the order of the data displayed.
    - Use the **Limit** field to specify the number of rows in a table or lines in a chart: up to 32 lines and 100 rows.
  - If you're creating a line chart, PEM supports comparisons of cross-hierarchy metrics.
    - Select **Compare** to select one or more probe-specific attributes (such as CPUs, interfaces, and databases) to compare in the chart.
    - Select **Copy** to apply your selections to all of the metrics for the same probe.

After you complete the fields on the Select Metrics dialog box, select **Next**.

3. Use the Set Options dialog box to specify display options for your chart:
  - Use the **Auto Refresh** field to specify the number of minutes between chart updates. Choose a value from 1 to 120. The default auto refresh rate is 2 minutes.

- Use fields under **Line chart options** to specify display preferences for a line chart:
  - Use the **Points to plot** field to specify the maximum number of points to plot on the chart.
  - Use the fields to the right of **Historical span** to specify how much historical data to display on the chart in days, hours, and minutes.
- Use the fields in the **Data extrapolation** box to specify whether PEM generates extrapolated data based on historical data:
  - Select **No Extrapolation** to omit extrapolated data from the chart.
  - Select **Span** to use the **Days** and **Hours** selectors to specify the period of time spanned by the metrics on the chart.
  - Select **Threshold** to use threshold selectors to specify a maximum or minimum value for the chart.

After you complete the fields in the Set Options dialog box, select **Next**.

4. Use the Set Permissions dialog box to specify display options for your chart.
  - Set the **Share with all** slider to **Yes** to make the chart available to all authorized users. Select **No** to restrict access to the users or groups specified in the **Access permissions** field.
  - Use the **Access permissions** field to select the groups you want to give access to the chart to.

After you finish defining the chart, select **Finish** to save your edits and add your chart to the list on the **Manage Charts** tab.

### Importing a Capacity Manager template

Select **Import Capacity Manager Template** in the Quick Links section of the **Manage Charts** tab to open the Create Chart dialog box and use a Capacity Manager template as a starting point for a chart or table.

1. When the Create Chart dialog box opens, provide information about the custom chart:
  - Use the **Import capacity template** list to select the name of the template on which the chart is based.
  - Specify the name of the chart in the **Name** field.
  - Use the **Category** list to specify the category in which to display this chart. When adding a custom chart to a custom dashboard, you can select the chart from this category.
  - Use the **Type** field to specify if the chart is a line chart or a table.
  - Provide a description of the chart in the **Description** field. The description is displayed on the custom dashboard when user selects the information icon next to the chart.

Select **Next**.

2. The Select Metrics window allows you to review the metrics specified by the selected template. The bottom panel of the chart editor displays the metrics to include in the chart. You can't modify the metrics included in the chart using the chart editor. To modify the metrics, use the Capacity Manager utility to update the template.

After you review the metrics, select **Next**.

3. Use the Set Options window to specify display options for your chart:
  - Use the **Auto Refresh** field to specify the number of minutes between chart updates. Choose a value from 1 to 120. The default auto refresh rate is 2 minutes.
  - Use the **Data extrapolation** box to specify the time period covered by the chart. You can either:
    - Select **Historical days and extrapolated days** and provide:
      - The number of days of historical data to chart in the **Historical** field.
      - The number of projected days to chart in the **Extrapolated** field.
    - Select **Historical days and threshold** and provide:

- The number of days of historical data to chart in the **Historical** field.
- The threshold value at which the chart ends.

After you complete the Set Options window, select **Next**.

4. Use the Set Permissions window to specify display options for your chart:

- Set the **Share with all slider** to **Yes** to make the chart available to all authorized users. Set it to **No** to restrict access to the users or groups specified in the **Access permissions** field.
- Use the **Access permissions** field to select the groups to give access to the chart to.

After you finish defining the chart, select **Finish** to save your edits and add your chart to the list on the **Manage Charts** tab.

## 22.8 Dashboards

PEM displays performance statistics through a number of dashboards. Each dashboard contains a series of summary views that contain charts, graphs, and tables that display the statistics related to the selected object.

### Dashboards overview

The PEM client displays the Global Overview dashboard when it connects to the PEM server. Additional dashboards provide statistical information about monitored objects.

### Opening dashboards

You can open a dashboard using either of these techniques:

- Select an active dashboard name from the **Management > Dashboards** menu.
- Right-click the name of a monitored object in the tree and select the name of the dashboard to review from the **Dashboards** menu.

Each dashboard is displayed on the **Monitoring** tab in the main panel of the client window. After opening a dashboard, you can navigate to other dashboards in the same tab.

Each dashboard header includes navigation menus that allow you to navigate to other dashboards. Use your browser's forward and back icons to scroll through previously viewed dashboards. Use **Refresh** to update the current dashboard.

### Alerts dashboard

The Alerts dashboard displays the currently triggered alerts. If opened from the Global Overview, the dashboard displays the current alerts for all monitored nodes on the system. If opened from a node in a server, the report shows alerts related to that node and all monitored objects that reside below that object in the tree.

### Audit Log Analysis dashboard

For EDB Postgres Advanced Server users, the Audit Log Analysis dashboard allows you to browse the audit logs that were collected from instances with audit logging and collection enabled.

**PGD Admin dashboard**

The PGD Admin dashboard displays overview information about the EDB Postgres Distributed node, group, and worker.

**PGD Group Monitoring dashboard**

The PGD Group Monitoring dashboard displays information about EDB Postgres Distributed group subscription and group replication slots.

**PGD Node Monitoring dashboard**

The PGD Node Monitoring dashboard displays information about EDB Postgres Distributed node slots, node replication rates, and conflict history summary for the selected node.

**Database Analysis dashboard**

The Database Analysis dashboard displays performance statistics for the selected database.

**I/O Analysis dashboard**

The I/O Analysis dashboard displays I/O activity across various areas such as object DML activity and log operations.

**Memory Analysis dashboard**

The Memory Analysis dashboard supplies statistics concerning various memory-related metrics for the Postgres server.

**Object Activity Analysis dashboard**

The Object Activity Analysis dashboard provides performance details on tables/indexes of a selected database.

**Operating System Analysis dashboard**

The Operating System Analysis dashboard supplies information regarding the performance of the underlying machine's operating system.

**Probe Log Analysis dashboard**

The Probe Log Analysis dashboard displays any error messages returned by a PEM agent.

**Server Analysis dashboard**

The Server Analysis dashboard provides general performance information about the overall operations of a selected Postgres server.

## Server Log Analysis dashboard

The Server Log Analysis dashboard allows you to filter and review the contents of server logs that are stored on the PEM server.

## Session Activity Analysis dashboard

The Session Activity Analysis dashboard provides information about the session workload and lock activity for the selected server

## Session Waits Analysis dashboard

The Session Waits Analysis dashboard provides an overview of the current DRITA wait events for an Advanced Server session.

## Storage Analysis dashboard

The Storage Analysis dashboard displays space-related metrics for tablespaces and objects.

## System Waits Analysis dashboard

The System Waits Analysis dashboard displays a graphical analysis of system wait information for an EDB Postgres Advanced Server session.

## Streaming Replication Analysis dashboard

The Streaming Replication Analysis dashboard displays statistical information about WAL activity for a monitored server and allows you to monitor the status of Failover Manager clusters.

## Dashboard configuration

Options on the Dashboard Configuration dialog box allow you to link the timelines of all of the line graphs on the dashboard. To open the Dashboard Configuration dialog box, select **Settings** in the dashboard header.

Use the Dashboard Configuration dialog box to control attributes of the charts displayed on the dashboard:

- Set **Link timelines of all the line charts** to **Enable** to apply the specified timeline to line graphs displayed on the dashboard. If set to **Disable**, your preferences are preserved for later use but don't modify the amount of data displayed.
- Use the **Days** selector to specify the number of days of gathered data to display on line graphs.
- Use the **Hour(s)** selector to specify the number of hours of gathered data to display on line graphs.
- Select **Remember configuration for this dashboard** to apply the customized time span to only the current dashboard only. Leave it cleared to apply the time span globally to line graphs on all dashboards.

Settings specified on the Dashboard Configuration dialog box are applied only to the current user's session.

## Managing custom dashboards

PEM displays performance statistics through a number of system-defined dashboards. Each dashboard contains a series of summary views that contain



charts, graphs, and tables that display statistics related to the selected object. You can use the **Manage Dashboards** tab to create and manage custom dashboards that display the information that's most relevant to your system.

To create a custom dashboard, select **Create New Dashboard** located in the Quick Links section of the **Manage Dashboards** tab.

To modify an existing dashboard, select **Edit** next to the dashboard name. The dashboard editor opens, displaying the definition of the dashboard. After you finish modifying the dashboard's definition, select **Save**.

To delete a dashboard, select **Delete** next to the dashboard name.

## Creating a custom dashboard

You can use the PEM dashboard editor to create or modify a custom dashboard. The custom dashboard can include predefined charts, custom charts, or a mix of predefined and custom charts.

Use the **Configure** section to specify general information about the dashboard:

- Specify a name for the dashboard in the **Name** field.
- Use the **Level** list to specify the level of the PEM hierarchy in the PEM client on which to display the dashboard. You can access a dashboard using the **Dashboards** menu on a global level, an agent level, the server level, or the database level. Each selected level in the list exposes a different set of metrics on which to base the custom dashboard's charts.
- Provide a description of the dashboard in the **Description** field.

Provide information in the **Ops dashboard options** box to use the dashboard as an Ops dashboard:

- Set **Ops Dashboard?** to **Yes** to create a dashboard that's formatted to display on an Ops monitor.
- Set **Show Title?** to **Yes** to display the dashboard name at the top of the Ops dashboard.
- Use the **Font** list to select a custom font style for the title.
- Use the **Font size** list to select a custom font size for the title.

Use the **Permissions** box to specify the users who can view the new dashboard:

- Set **Share with all** to **Yes** to allow all teams to access the dashboard. Set it to **No** to enable the **Access permissions** field.
- Use the **Access permissions** field to specify the roles that can view the new dashboard. In the field, select from the list of users to add a role to the list of users with dashboard access.

After you complete the **Configure Dashboard** section, select the arrow in the upper-right corner to close the section and access the **Dashboard Layout Design** section.

Select **Edit** in a section header to specify a section name. Then, select the plus sign (+) to add a chart to the section.

Use the arrows to the right of each chart category to display the charts available and select a chart.

Use the chart detail selectors to specify placement details for the chart:

- Use the **Chart width** selector to indicate the width of the chart. Select **50%** to display the chart in half of the dashboard or **100%** to use the whole dashboard width.
- Use the **Chart alignment** selector to indicate the position of the chart in the section, left, center, or right justified.

Tables are always displayed centered.

When creating or editing a custom dashboard, you can drag to rearrange the charts in a section or to move a chart to a different section.

To add another chart to your dashboard, select the plus sign (+) in the section header. After you finish editing the dashboard, select **Save**.

Creating an Ops dashboard

You can use the PEM dashboard editor to create a custom dashboard formatted for display on an Ops monitor. An Ops dashboard displays the specified charts and graphs while omitting header information and minimizing extra banners, titles, and borders.

To create an Ops dashboard, provide detailed information about the Ops display in the **Ops dashboard options** section of the Create Dashboard dialog box.

- Set **Ops Dashboard?** to **Yes** to create a dashboard that's formatted for display on an Ops monitor.
- Set **Show Title?** to **Yes** to display the dashboard name at the top of the Ops dashboard.
- Use the **Font** list to select a custom font style for the title.
- Use the **Font size** list to select a custom font size for the title.

After adding charts and tables to the Ops dashboard, select **Save**. You can then access the dashboard by navigating through the **Dashboards** menu of the hierarchy level specified in the **Level** field on the New Dashboard dialog box.

22.9 Remote monitoring

Remote monitoring is monitoring your Postgres cluster using a PEM Agent residing on a different host.

To remotely monitor a Postgres cluster with PEM, you must register the cluster with PEM and bind a PEM agent. See [Registering a server](#) for more information.

The following scenarios require remote monitoring using PEM:

- Postgres cluster running on AWS RDS
- [Postgres cluster running on BigAnimal](#)

PEM remote monitoring supports:

| Feature Name                                 | Remote monitoring supported? | Comments  |
|--|------------------------------|---|
| <a href="#">Manage charts</a>                | Yes                          |   |
| <a href="#">System reports</a>               | Yes                          |   |
| <a href="#">Capacity Manager</a>             | Limited                      | There's no correlation between the Postgres cluster and operating system metrics.   |
| <a href="#">Manage alerts</a>                | Limited                      | When you run an alert script on the Postges cluster, it runs on the machine where the bound PEM agent is running and not on the actual Postgres cluster machine.                    |
| <a href="#">Manage dashboards</a>            | Limited                      | Some dashboards might not be able to show complete data. For example, the operating system information where the Postgres cluster is running isn't displayed as it isn't available. |
| <a href="#">Manage probes</a>                | Limited                      | Some of the PEM probes don't return information, and some of the functionality might be affected. For details about probe functionality, see <a href="#">PEM agent privileges</a> . |
| <a href="#">Postgres Expert</a>              | Limited                      | The Postgres Expert provides partial information as operating system information isn't available.   |
| <a href="#">Scheduled tasks</a>              | Limited                      | Scheduled tasks work only for Postgres clusters, and scripts run on a remote agent.   |
| <a href="#">Core usage reports</a>           | Limited                      | The Core Usage reports don't show complete information. For example, the platform, number of cores, and total RAM aren't displayed.   |
| <a href="#">Audit manager</a>                | No                           |   |
| <a href="#">Log manager</a>                  | No                           |   |
| <a href="#">Postgres Log Analysis Expert</a> | No                           |   |
| <a href="#">Tuning wizard</a>                | No                           |   |

## 23 Monitoring Barman

### New Feature

Monitoring Barman is available in PEM 8.4.0 and later.

Barman (Backup and Recovery Manager) is an open-source administration tool for remote backups and disaster recovery of PostgreSQL servers in business-critical environments. It relies on PostgreSQL's point-in-time recovery technology, allowing DBAs to remotely manage a complete catalog of backups and the recovery phase of multiple remote servers from one location. For more information, see [Barman](#).

Starting with version 8.4, you can monitor a Barman server through the PEM console.

### Prerequisites for monitoring Barman

Before adding a Barman server to the PEM console:

- You must manually install and configure Barman on the Barman host. For more information about installing and configuring Barman, see [Barman](#).
- Install the pg-backup-api tool on Barman host. For more information about installing, see [pg-backup-api](#).

### Configuring a Barman server

You can configure and edit your Barman server using:

- PEM web client
- `pemworker` command line

#### Using PEM web client

##### Configure

You can use the Create-BARMAN Server dialog box to register an existing Barman server with the PEM server. To open the dialog box, right-click the BARMAN Servers node and select **Create-BARMAN Server**.

Use the **General** tab to describe the general properties of the Barman server:

- Use the **Name** field to specify a name for the server. The name identifies the server in the browser tree.
- Use the **URL** field to specify the URL of the host where Barman is installed.
- Use the **Team** field to specify a PostgreSQL role name. Only PEM users who are members of this role, who created the server initially, or have superuser privileges on the PEM server can see this server when they log on to PEM. If this field is left blank, all PEM users see the server.

Use the **PEM Agent** tab to specify connection details for the PEM Agent:

- Use the **Bound Agent** field to select the agent that you want to configure as a Barman server. Only those PEM agents that are supported for Barman are listed.

- Use the **Probe Frequency** field to specify the number of seconds to execute the probes with the specified interval.
- Use the **Heartbeat** field to specify the interval to check the availability of PEM agent in seconds.

### Note

After registering the Barman server, you need to restart the PEM agent.

### Editing

To edit your Barman server, right-click the server from the browser tree and select **Properties**.

- Use the **PEM Agent** tab to modify the bound agent, probe frequency, and heartbeat. Only the owner of the Barman server can modify the fields on the **PEM Agent** tab.
- Use the **Information** tab to view the detailed information about your Barman server. This tab gets populated whenever the Barman related probes are executed.
- Use the **Configuration** tab to view the configuration settings of your Barman server. This tab gets populated whenever the Barman related probes are executed.

### Using pemworker command line

You can configure Barman server using `pemworker` command line options.

```
asheshvashi@pem:~/PEM/agent$ ./pemworker --update-barman --help
./pemworker --update-barman [barman-update-options]

barman-update-options:
  --id <barman-id> (ID for the existing BARMAN API 'pg-backup-api')
  --api-url <url> (URL of the BARMAN API 'pg-backup-api')
  --probe-execution-frequency <interval> (Default: 30, Probe the BARMAN API 'pg-backup-api' at regular
interval 'in seconds' and fetch the metrics.)
  --heartbeat-interval <interval> (Default: 10, Ping the BARMAN API 'pg-backup-api' 'status' API at a
regular interval 'in seconds' for checking its availability.)
  --ssl-crt <certificate_file> (SSL certificate file for the BARMAN API.)
  --ssl-key <key_file> (Private SSL key for the BARMAN API.)
  --ssl-ca-cert <ca_file> (CA certificate to verify peer against the BARMAN API.)
  --config-file/-c <config_file> (Path to the agent configuration file.)

asheshvashi@pem:~/PEM/agent$ ./pemworker --unregister-barman --help
./pemworker --unregister-barman [barman-unregistration-options]

barman-unregistration-options:
  --id <barman-id> (ID for the existing BARMAN API, registered with the PEM Server.'pg-backup-api')
  --config-file/-c <config_file> (Path to the agent configuration file.)

asheshvashi@pem:~/PEM/agent$ ./pemworker --register-barman --help
./pemworker --register-barman [barman-registration-options]

barman-registration-options:
  --api-url <url> (URL of the BARMAN API 'pg-backup-api')
  --description <name> (Description to show on the UI 'User interface' for the BARMAN API.)
  --probe-execution-frequency <interval> (Default: 30, Probe the BARMAN API 'pg-backup-api' at regular
interval 'in seconds' and fetch the metrics.)
  --heartbeat-interval <interval> (Default: 10, Ping the BARMAN API 'pg-backup-api' 'status' API at a
```

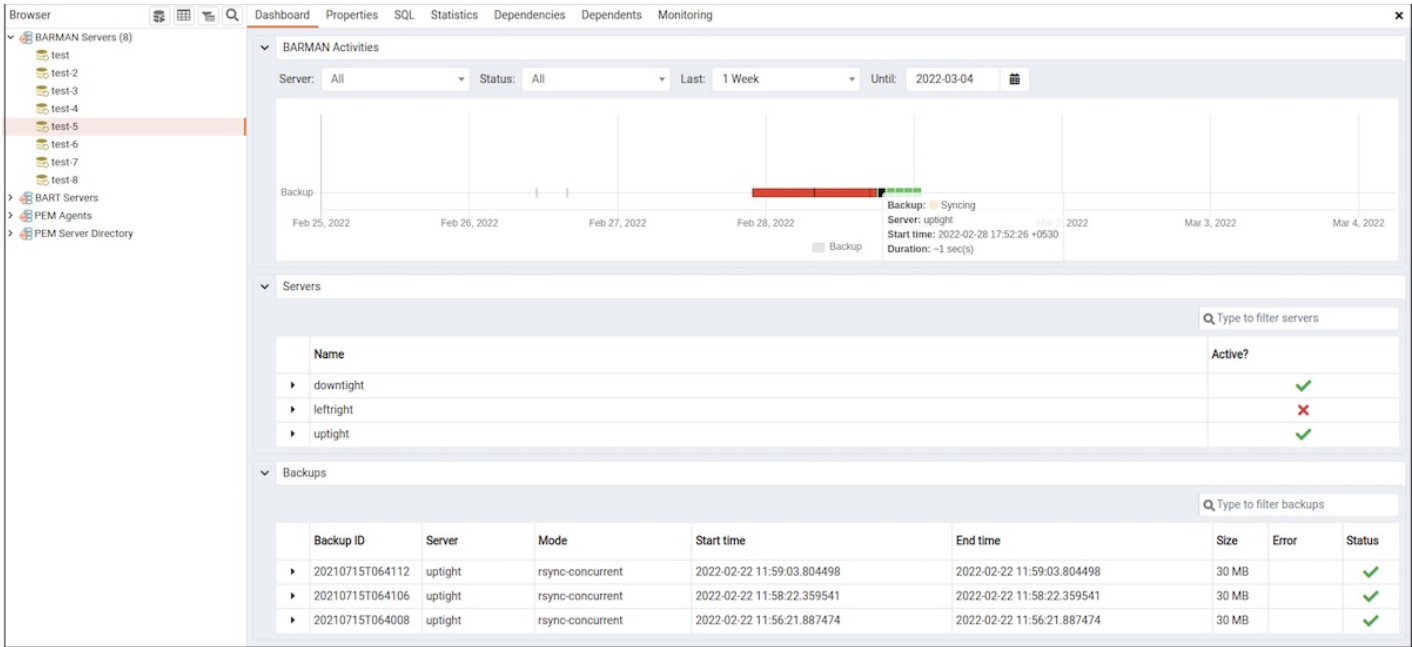
```
regular interval 'in seconds' for checking its availability.)
--ssl-crt <certificate_file> (SSL certificate file for the BARMAN API.)
--ssl-key <key_file> (Private SSL key for the BARMAN API.)
--ssl-ca-cert <ca_file> (CA certificate to verify peer against the BARMAN API.)
--team <database-role> (Specify the name of the database group role, on the PEM backend database
server, that should have access to this BARMAN API Server.)
--owner <database-user> (Specify the name of the database user, on the PEM backend database server,
who will own the BARMAN API Server.)
--config-file/-c <config_file> (Path to the agent configuration file.)
```

Note

After registering the Barman server, you need to restart the PEM agent.

Viewing the Barman server details on a PEM dashboard

Once the Barman server is configured, you can see the entire backup- and server-related details for that Barman server on the PEM dashboard.



When you select a monitored Barman server, details of all the associated database servers along with their activities are displayed as a chart on the dashboard in the Barman Activities panel. You can select the activities on any criteria that you specify in the filter boxes (the database server, status, duration, or date).

The Servers panel displays a list of all the database servers managed by that Barman server along with the active status.

The Backups panel displays a list of all the database server backups managed by that Barman server. You can filter the list to display the details of any database server. You can also filter the list on any criteria that you specify in the filter box. Typically, this filter works with any kind of string value (excluding date, time, and size) listed under the columns. For example, you can enter `tar` to filter the list and display only those backups that are in tar format.

Backup details include the Backup ID, Server, Mode, Start time, End time, Size, Error, and Status column.

24 Monitoring EDB Postgres Distributed

**\*\*New Feature \*\***

EDB Postgres Distributed support is available in PEM version 8.1.0 and later.

EDB Postgres Distributed provides multi-master replication and data distribution with advanced conflict management, data-loss protection, and [throughput up to 5X faster than native logical replication](#), and enables distributed PostgreSQL clusters with high availability up to five 9s. Before you monitor nodes in a EDB Postgres Distributed cluster through the PEM console, you must first deploy a EDB Postgres Distributed cluster and ensure that your database nodes are up and running. For more information on installing EDB Postgres Distributed see [EDB Postgres Distributed](#).

You can configure PEM to display status information about one or more EDB Postgres Distributed database nodes using dashboards in PEM version 8.1.0 and EDB Postgres Distributed version 3.7.9 and later.

To configure PEM to monitor EDB Postgres Distributed database nodes, use the PEM web client to create a server definition. Use the tabs on the [New Server Registration](#) dialog box to specify general connection properties for the EDB Postgres Distributed database node with the following exceptions:

- Specify the EDB Postgres Distributed-enabled database name in the **Database** field of the **PEM Agent** tab.
- Specify the user having the pgd\_monitor or pgd\_superuser role in the **username** field of the **PEM Agent** tab.

After saving the server definition, the EDB Postgres Distributed database node is included in the list of servers under the PEM server directory in the PEM client object browser tree. You can monitor the nodes from EDB Postgres Distributed (PGD) dashboards.

To include monitoring information on the EDB Postgres Distributed (PGD) dashboards, you must enable the relative probes for each EDB Postgres Distributed group. See the complete list of EDB Postgres Distributed [probes](#).

To enable a probe, right-click the node name, and select **Management > Manage Probes**.

To monitor the EDB Postgres Distributed database node, right-click the name of the node in the object browser tree. From the **Dashboards** menu, select the PGD Admin, PGD Group Monitoring, or PGD Node Monitoring dashboard.

## 25 Monitoring Failover Manager

If you're using EDB Failover Manager to monitor your replication scenario, you must manually install and configure Failover Manager. For detailed information about installing Failover Manager, see the [EDB website](#).

To monitor the status of a Failover Manager cluster on the Streaming Replication dashboard, you must provide the following information on the **Advanced** tab of the **Server Properties** dialog box for each node of the cluster:

- Use the **EFM Cluster Name** field to specify the name of the Failover Manager cluster. The cluster name is the prefix of the name of the cluster properties file. For example, if your cluster properties file is named `efm.properties`, your cluster name is efm.
- Use the **EFM Installation Path** field to specify the location of the Failover Manager binary file. By default, the Failover Manager binary file is installed in `/usr/efm-<X>/bin`, where `<X>` is the EFM version.

After registering your servers, the Streaming Replication Analysis dashboard displays status information about your EFM cluster near the bottom of the dashboard.

Failover Manager Cluster Status

Failover Manager Cluster Information

| Properties                            | Values                        |
|---------------------------------------|-------------------------------|
| Cluster Name                          | efm                           |
| Failover Manager Agent Running Status | UP                            |
| Allowed Node List                     | 172.16.177.194, 172.16.23.156 |
| Standby Priority List                 | 172.16.23.156                 |
| Missing Nodes                         |                               |
| Minimum Standbys                      | 0                             |
| Membership Coordinator                | 172.16.177.194                |
| Cluster Status Message                |                               |

Failover Manager Node Status

| Agent Type | Address        | Agent | DB | XLog Location | XLog Receive | Status Information | XLog information | VIP | VIP Status |
|------------|----------------|-------|----|---------------|--------------|--------------------|------------------|-----|------------|
| Primary    | 172.16.177.194 | UP    | UP | 0/8000140     |              |                    |                  |     | False      |
| Standby    | 172.16.23.156  | UP    | UP | 0/8000140     | 0/8000140    |                    |                  |     | False      |

The **Failover Manager Cluster Status** section of the Streaming Replication Analysis dashboard displays information about the monitored cluster.

The Failover Manager Cluster Information table provides information about the Failover Manager cluster:

- The Properties column displays the name of the cluster property.
- The Values column displays the current value of the property.

The Failover Manager Node Status table displays information about each node of the Failover Manager cluster:

- The Agent Type column displays the type of agent that resides on the node. The possible values are Primary, Replica, Witness, Idle, and Promoting.
- The Address column displays the IP address of the node.
- The Agent column displays the status of the agent that resides on the node.
- The DB column displays the status of the database that resides on the node.
- The XLog Location column displays the transaction log location of the database.
- The Status Information column displays any error-related information about the node.
- The XLog Information column displays any error-related information about the transaction log.
- The VIP column displays the VIP address that's associated with the node.
- The VIP Status column displays True if the VIP is active for the node, False if not.

## Replacing a primary node

You can use the PEM client to replace the primary node of a Failover Manager cluster with a replica node. To start the failover process, select **Tools > Server > Replace Cluster Primary**. You are prompted to confirm that you want to replace the current primary node.

Select **Yes** to remove the current primary node from the Failover Manager cluster and promote a replica node to the role of read/write primary node in a Failover Manager cluster. The node with the highest promotion priority (defined in Failover Manager) becomes the new primary node. PEM reports the job status.

When the job completes and the Streaming Replication Analysis dashboard refreshes, you can review the Failover Manager Node Status table to confirm that a replica node was promoted to the role of primary in the Failover Manager cluster.

## Switchover EFM cluster

You can use the PEM client to replace the primary node of a Failover Manager cluster with a replica node. To start the switchover process, select **Tools > Switchover EFM Cluster**. You are prompted to confirm that you want to switch over EFM cluster. Select **Yes** to:

- Start the Failover Manager switchover.
- Promote a replica node to the role of read/write primary node.
- Reconfigure the primary database as a new replica in a Failover Manager cluster.

The node with the highest promotion priority (defined in Failover Manager) becomes the new primary node. PEM reports the job status. When the job completes and the Streaming Replication Analysis dashboard refreshes, you can review the Failover Manager Node Status table to confirm that a switchover occurred.

## 26 Monitoring Replication Server

Before configuring PEM to retrieve statistics from an EDB Postgres Advanced Server or PostgreSQL database that's part of an xDB replication scenario, you must manually install and configure xDB replication. For more information about xDB replication solutions and documentation, see [Multi-master replication operation](#).

The PEM xDB Replication probe monitors lag data for clusters that use xDB multi-primary or single-primary replication that have an EDB Postgres Advanced Server or PostgreSQL publication database. If you configured replication between other proprietary database hosts (that is, Oracle or SQL Server) and EDB Postgres Advanced Server or PostgreSQL, the probe can't return lag information.

By default, the xDB Replication probe is disabled. To enable the xDB Replication probe:

1. Right-click the name of the server and select **Connect** from the context menu.
2. If prompted, provide authentication information.
3. After connecting, expand the server node of the tree and select the name of the replicated database.
4. Select **Management > Manage Probes**.

Use the **Manage Probes** tab to configure the xDB Replication probe:

- Set **Default** to **No** to modify the minutes and seconds between probe executions.
- Use the **Enabled?** slider to enable the xDB Replication probe.
- Set **Default** slider in the **Data Retention** field to **No** to modify the number of days for PEM to store the information retrieved by the probe.

After enabling the probe, you can use the metrics returned to create custom charts and dashboards in the PEM client.

## 27 Tuning performance

The following features help in tuning the performance:

- [Tuning Wizard](#)
- [Postgres Expert](#)
- [Performance Diagnostics](#)

### 27.1 Tuning Wizard

The Tuning Wizard reviews your PostgreSQL or EDB Postgres Advanced Server installation, recommending a set of configuration options tuned to the installation's anticipated workload. Benchmarking systems or systems with a high work load might require also require manual tuning to reach optimum performance.

Before using the Tuning Wizard, you must specify the name of the service in the server's Properties dialog box. In the **Advanced** tab, enter the service name in the **Server ID** field.

The Tuning Wizard can make recommendations only for those servers that reside on the same server as their bound PEM agent. If you specify a value of **Yes** in the **Remote monitoring** field when defining your server, the server doesn't display in the Tuning Wizard tree.

1. To open the Tuning Wizard, in the PEM client select **Management > Tuning Wizard**.



2. Select **Next**.

3. When you expand the **Servers** node of the tree, a list of servers appears. All of these servers are currently monitored by PEM and available for tuning. Select a server to tune it.

#### Note

If you don't provide the server's service name, then the Tuning Wizard displays a warning next to the server name on the tree.

Select **Next**.

4. Select an option in the **Machine utilization** field to specify the type of work performed by each server. The type of work performed by the server determines how the Tuning Wizard allocates system resources:

- Select **Dedicated** to dedicate the majority of the system resources to the database server.
- Select **Mixed use** to dedicate a moderate amount of system resources to the database server.
- Select **Developer workstation** to dedicate a relatively small amount of system resources to the database server.

Select an option in the **Workload Selection** field to specify the type of workload typically performed on the selected server:

- Select **OLTP** if the selected server is used primarily to process online transaction workloads.
- Select **Mixed** if the selected server provides a mix of transaction processing and data reporting.
- Select **Data warehouse** if the server is used for heavy data reporting.

Select **Next**.

1. The tree on the Tuning Changes Summary dialog box displays the parameter setting modifications recommended for each server analyzed by the Tuning Wizard. Select the recommendations that you want the Tuning Wizard to apply or to include in a preview report:

- Select a parameter name and the Tuning Wizard includes the parameter setting.
- Select the server name and the Tuning Wizard includes all parameter setting recommendations for the specified server.

Select **Next**.

1. In the **Schedule or Run?** dialog box, either select a time for PEM to apply the recommended changes or generate a report that details the recommended changes.

PEM makes the recommended changes that you selected in the **Tuning Changes Summary** dialog box. If you choose to generate a report, then PEM creates a report. It contains a list of the current values and recommended changes to the configuration parameters as selected in the **Tuning Changes Summary** dialog box. To implement changes, you must open the Tuning Wizard a second time, selecting the parameters you want to modify in the **Tuning Changes Summary** dialog box.

Select **Schedule changes** to view and specify your scheduling options.

You can set the **Configuration now?** slider to:

- **Yes** — Apply the Tuning Wizard's recommendations and restart the server.
- **No** — Enable the **Time?** field where you can specify a date and time with the calendar selector. PEM applies the recommended changes and restarts the server at this time.

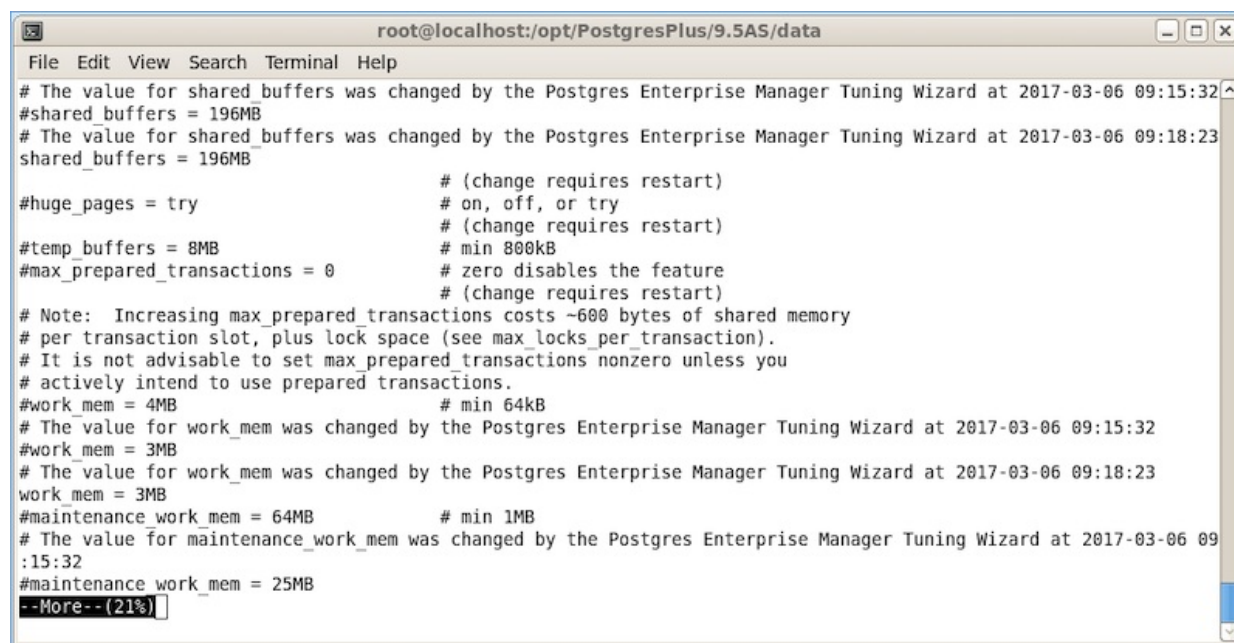
Select **Generate report** to view your report options.

You can set the **View report now?** slider to:

- **Yes** — Display the Tuning Wizard report onscreen.
- **No** — Enable the **Save the report to file** field where you can specify a file name and location.

1. Select **Finish**.

To confirm that the Tuning Wizard implemented the recommended changes, review the `postgresql.conf` file for the modified server. When the change is applied, the Tuning Wizard adds a comment above each modified parameter in the `postgresql.conf` file.



```

root@localhost:/opt/PostgresPlus/9.5AS/data
File Edit View Search Terminal Help
# The value for shared_buffers was changed by the Postgres Enterprise Manager Tuning Wizard at 2017-03-06 09:15:32
#shared_buffers = 196MB
# The value for shared_buffers was changed by the Postgres Enterprise Manager Tuning Wizard at 2017-03-06 09:18:23
shared_buffers = 196MB

# (change requires restart)
# on, off, or try
# (change requires restart)
# min 800kB
# zero disables the feature
# (change requires restart)
# Note: Increasing max_prepared_transactions costs ~600 bytes of shared memory
# per transaction slot, plus lock space (see max_locks_per_transaction).
# It is not advisable to set max_prepared_transactions nonzero unless you
# actively intend to use prepared transactions.
#work_mem = 4MB # min 64kB
# The value for work_mem was changed by the Postgres Enterprise Manager Tuning Wizard at 2017-03-06 09:15:32
#work_mem = 3MB
# The value for work_mem was changed by the Postgres Enterprise Manager Tuning Wizard at 2017-03-06 09:18:23
work_mem = 3MB
#maintenance_work_mem = 64MB # min 1MB
# The value for maintenance_work_mem was changed by the Postgres Enterprise Manager Tuning Wizard at 2017-03-06 09:15:32
#maintenance_work_mem = 25MB
--More-- (21%)

```

You can also confirm a parameter value by querying the server. For example, to confirm the value of the `shared_buffers` parameter, open a SQL command line using either the Query tool (accessed through the Tools menu) or the `psql` client, and issue the command:

```
SHOW shared_buffers;
```

The value returned by the server confirms whether the parameter was modified.

## 27.2 Postgres Expert

The Postgres Expert utility provides expert advice on how to best configure your Postgres servers for optimal performance, security, and more. Postgres Expert serves as a PostgreSQL DBA-in-a-box by evaluating your servers for deviations in best practices. Postgres Expert contains three specialized experts:

- Configuration Expert
- Schema Expert
- Security Expert

You can select specific rules for each expert to evaluate or accept all rules. Then, review Postgres Expert reports detailing any best practice issues that require your attention.

### Using the Postgres Expert wizard

To use the Postgres Expert wizard, in the PEM client select **Management > Postgres Expert**.

1. In the Welcome message, select **Next**.
2. The **Experts/Rules** tree lists the available experts and rules for identifying best practice deviations. Select the ones you want to use to evaluate the selected servers or databases.

The tree categorizes the rules under three experts:

- **Configuration Expert** evaluates the parameter settings of the server or operating system to find any adjustments that might improve system performance.
- **Schema Expert** evaluates schema objects, such as locating missing primary keys or foreign keys without indexes.
- **Security Expert** evaluates the system to find security vulnerabilities.

After making your selections, select **Next**.

1. Select or clear the servers and databases that you want to evaluate. If you select multiple servers or databases, then the resulting report contains a separate evaluation of each target. After you finish, select **Next**.
2. To view the report in the client, select **View the report now**. To save a copy to an HTML file, select **Download the report**. If you download the report, then the file saves to your default download directory. Select **Finish**.

## Reviewing Postgres Expert recommendations

Postgres Expert produces a report that contains an evaluation of the selected rules. Each rule is categorized as high, medium, or low severity for the selected servers.

The report header contains a summary of the report that includes:

- The date and time of the report
- The number of rules evaluated
- The number of deviations from best practices

To jump to the section for a given server, select the server from the **Go to** list.

The body of the report contains the detailed findings for each of the selected servers. The findings are sorted by expert. Under each heading, any rule violations are ranked by severity.

Postgres Expert Report

Generated On: 2020-04-27 16:57:47

Go to: Postgres Enterprise Manager Server

Summary

Servers Tested: 1 Rules Checked: 31 High Alerts: 1 Medium Alerts: 3 Low Alerts: 2

Server: Postgres Enterprise Manager Server (192.168.1.18:5432)

Advisor: Configuration Expert

| Rule                               | Database | Severity |
|------------------------------------|----------|----------|
| Check checkpoint_completion_target | -        | Medium   |

Recommended Value: Consider adjusting checkpoint\_completion\_target.

Current Values:

| Settings                     | Value |
|------------------------------|-------|
| checkpoint_completion_target | 0.5   |

Trigger: checkpoint\_completion\_target != 0.9

Description: In order to ensure reliable and efficient crash recovery, PostgreSQL periodically writes all dirty buffers to disk. This process is called a checkpoint. Beginning in PostgreSQL 8.3, checkpoints take place over an extended period of time in order to avoid swamping the I/O system. checkpoint\_completion\_target controls the rate at which the checkpoint is performed, as a function of the time remaining before the next checkpoint is due to start. A value of 0 indicates that the checkpoint should be performed as quickly as possible, whereas a value of 1 indicates that the checkpoint should complete just as the next checkpoint is scheduled to start. It is usually beneficial to spread the checkpoint out as much as possible; however, if checkpoint\_completion\_target is set to a value greater than 0.9, unexpected delays near the end of the checkpoint process can cause the checkpoint to fail to complete before the next one needs to start. Because of this, the recommended setting is 0.9.

Check effective\_cache\_size

-

Medium

Check effective\_io\_concurrency

-

Low

Check reducing\_random\_page\_cost

-

Low

Advisor: Schema Expert

| Rule   | Database | Severity |
|--|----------|----------|
| Check data and transaction log on same drive | -        | High     |
| Check for missing foreign key indexes        | db01     | Medium   |

Select each rule in the report to display details and recommendations for that rule. Selecting the rule displays these section headings:

- Advisor lists the name of the Postgres Expert advisor that prompted the recommendation.
- Trigger displays a description of the rule that raised the alert.
- Recommended Value displays the value to which Postgres Expert recommends setting the selected parameter.
- Description displays information and advice about the parameter that caused the alert.
- Current Values displays the current values of any parameters that influence the Postgres Expert evaluation.

### 27.3 Performance Diagnostic

The Performance Diagnostic dashboard analyzes the database performance for Postgres instances by monitoring the wait events. To display the diagnostic graphs, PEM uses the data collected by the EDB Wait States module.

Note

- For PostgreSQL databases, Performance Diagnostic is supported for version 10 or later installed on the supported RHEL platforms.
- For EDB Postgres Extended databases, Performance Diagnostic is supported for version 11 or later on the supported RHEL platforms.

For more information on EDB wait states, seeEDB wait states.

To analyze the Wait States data on multiple levels, narrow down the data you select. The data you select at the higher level of the graph populates the lower level.

Prerequisites

- For PostgreSQL, you need to install the `edb_wait_states_<X>` package from `edb.repo`, where `<X>` is the version of PostgreSQL Server. For the steps to install this package, see [EDB Build Repository](#). For EDB Postgres Advanced Server, you need to install the `edb-as<X>-server-edb-modules`, where `<X>` is the version of EDB Postgres Advanced Server.
- After you install the EDB Wait States module of EDB Postgres Advanced Server:

- Configure the list of libraries in the `postgresql.conf` file as shown:

```
shared_preload_libraries = '$libdir/edb_wait_states'
```

- Restart the database server.
- Create the following extension in the maintenance database:

```
CREATE EXTENSION
edb_wait_states;
```

- You need superuser privileges to access the Performance Diagnostic dashboard.

If the prerequisites aren't met, then an error appears when you access the Performance Diagnostic dashboard:

## Using the Performance Diagnostic dashboard

To open the Performance Diagnostic dashboard, on the PEM client select **Tools > Server > Performance Diagnostic**.



By default, the top Performance Diagnostic graph pulls the data of the last hour, starting from the current date and time. This graph shows the time series containing the number of active sessions. Each point of this time series represents the active sessions along with the wait events at a particular time and in the last 15 seconds. These sessions might be waiting for a wait event or using the CPU at a particular point in time. This time series is generated based on the wait event samples collected by the EDB Wait States extension.

To open the Performance Diagnostic dashboard in a new browser tab, select **Preferences > Open in New Browser Tab?**

The duration selection in the first graph is 10 minutes. To see the duration on a graph, select a duration from the **Last** list. To display the data for a specified date and time, select a date and time from the **Until** list.

The first graph displays the number of active sessions and wait event types for the selected duration. You can change the duration in the first graph to analyze the data for a specific time period.

The next section plots the following graphs based on the selected duration in the first graph:

**Donut graph** — Shows total wait event types according to the duration selection in the first graph. It can provide a better understanding of how much time was spent by those sessions on waiting for an event.

**Line graph** — Plots a time series with each point representing the active sessions for each sample time.

To differentiate each wait event type and the CPU usage more clearly, the graph for each wait event type displays in a different color.

Select a time on the Line graph to analyze the wait events. The third section displays the wait event details in the Performance Diagnostics dashboard based on your selected duration in the second graph. The third section displays wait event details on three tabs:

- The **SQL** tab displays the list of SQL queries having wait events for the selected sample time.
- The **Users** tab displays the details of the wait events grouped by users for the selected sample time.
- The **Waits** tab displays the number of wait events belonging to each wait event type for the selected sample time.

To show or hide a wait event type in all the graphs, select the graph legends. The analysis is simpler when you can only see the wait event types you want to analyze.

You can filter the data displayed in the rows under all three tabs. You can also sort the data alphabetically by selecting the column headers.

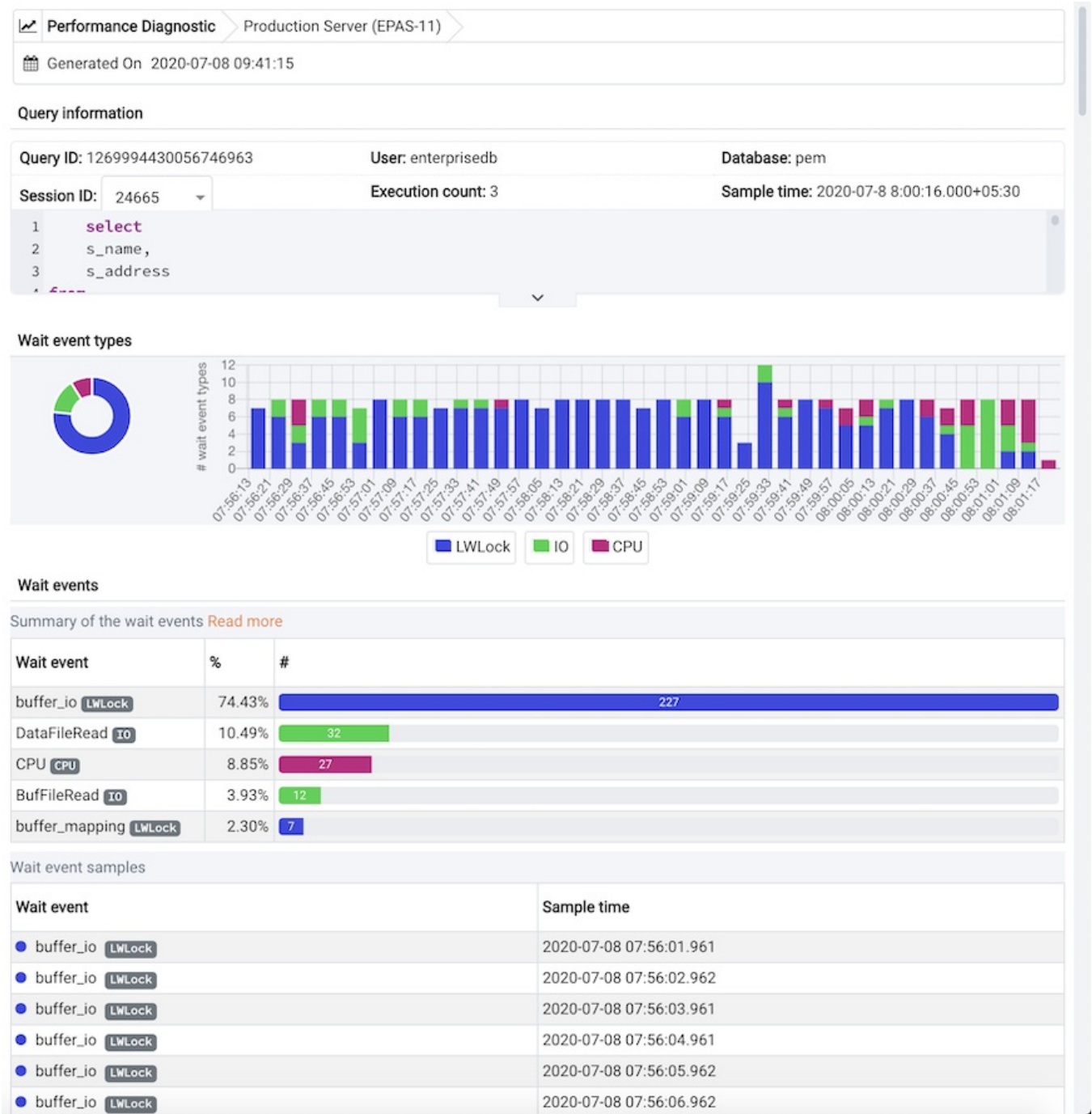
Select the eye in any row of the **SQL** tab to display a window with details on the query of that row. This window displays a query ID and its corresponding session IDs in a list at that selected sample time in the **Query information** section. You can select the **Session ID** list for the selected query that you want to analyze the data. The details corresponding to the selected session ID and query ID appear. The **Query information** table also displays the SQL query.

The **Wait event types** section displays the total number of wait event types for the selected session ID and query ID. It shows two types of graphs:

**Donut graph** — Shows the proportions of categorical data.

**Timeline bar graph** — Visualizes trends in counts of wait event types over time.

To differentiate, each wait event type is represented by a different color in the bar graph.



## 28.1 SQL Profiler

Postgres Enterprise Manager (PEM) is designed to assist database administrators, system architects, and performance analysts when administering, monitoring, and tuning PostgreSQL and EDB Postgres Advanced Server database servers. PEM was designed to manage and monitor a single server or multiple servers from a single console, allowing complete control over monitored databases.

The SQL Profiler Plugin works with PEM to allow you to profile a server's workload. You can install the SQL Profiler plugin on servers with or without a PEM Agent. However you can run traces only in ad hoc mode on unmanaged servers, and you can schedule them only on managed servers.

SQL Profiler is officially supported only on the EDB distributions of PostgreSQL and EDB Postgres Advanced Server supported versions. The plugin is distributed using StackBuilder or as operating system-dependent packages in EDB's yum repositories. The plugin is also distributed and installed with the EDB Postgres Advanced Server installations.

### 28.1.1 Installing the SQL Profiler plugin

You must install the plugin on each server on which you want to use SQL Profiler. For example, if you have a host running PostgreSQL 10 and PostgreSQL 11, you must install two versions of the plugin, one for each server.

Install the plugin for PostgreSQL before configuring it. If you're using EDB Postgres Advanced Server, you can skip installation and go to [Configuring SQL Profiler](#).

You can use the graphical installer to install any version of SQL Profiler on the Windows platform. On Linux, use an RPM package to install the SQL Profiler. For detailed information about configuring the EDB repository for your host platform, see the [Installing on the PEM server](#).

#### Installing SQL Profiler on Windows

1. To invoke the SQL Profiler graphical installer, assume Administrator privileges, navigate to the directory that contains the installer, and double-click the installer icon. The SQL Profiler installer welcomes you to the Setup Wizard.
2. Select **Next** to continue to the license agreement.
3. Review the license agreement before selecting the appropriate radio button and accepting the agreement. Select **Next**.
4. Specify an alternative location for the installation directory, or accept the default location and select **Next**.
5. The wizard is now ready to install the SQL Profiler plugin. Select **Next**.

When the installation is complete, the SQL Profiler plugin is ready to configure.

#### Installing SQL Profiler on RHEL

##### Note

You might need to add the `ssltutils` package to your PostgreSQL database servers before installing SQL Profiler.

If you already configured the EDB repository on your system, you can use `dnf` to install SQL Profiler.

For PostgreSQL,



```
dnf install sqlprofiler_<X>
```

Where `<X>` is the version of your Postgres server.

For EDB Postgres Advanced Server,

```
yum install edb-as<X>-server-sqlprofiler
```

or

```
dnf install edb-as<X>-server-sqlprofiler
```

Where `<X>` is the version of your EDB Postgres Advanced Server installation.

For detailed information about configuring the EDB repository, see [Installing the PEM server](#).

## Installing SQL Profiler on Debian/Ubuntu

### Note

You might need to add the `sslutils` package to your PostgreSQL database servers before installing SQL Profiler.

You can use an `apt` command to install SQL Profiler using DEB on Debian and Ubuntu. Assume root privileges.

For PostgreSQL,

```
apt install sqlprofiler_<X>
```

Where `<X>` is the version of your Postgres installation.

For EDB Postgres Advanced Server,

```
apt install edb-as<X>-server-sqlprofiler
```

Where `<X>` is the version of your EDB Postgres Advanced Server installation.

When the installation is complete, the SQL Profiler plugin is ready to configure.

## Installing SQL Profiler on SLES

### Note

You might need to add the `sslutils` package to your PostgreSQL database servers before installing SQL Profiler.

You can use an `zypper` command to install SQL Profiler on SLES. Assume root privileges.

For PostgreSQL,

```
zypper install sqlprofiler_<X>
```

Where `<X>` is the version of your Postgres server.

For EDB Postgres Advanced Server,

```
zypper install edb-as<X>-server-sqlprofiler
```

Where `<X>` is the version of your EDB Postgres Advanced Server installation.

When the installation is complete, the SQL Profiler plugin is ready to configure.

## Configuring SQL Profiler

The SQL Profiler plugin isn't enabled when the installation process completes. This way you can restart the server at a convenient time. It also prevents the plugin from being loaded on systems where you don't need it all the time.

### Note

If you connected to the PEM server with the PEM client before configuring SQL Profiler, you must disconnect and reconnect with the server to enable SQL Profiler functionality.

To enable the plugin:

1. Edit the `postgresql.conf` file on the server you want to profile, modifying the `shared_preload_libraries` parameter:

```
shared_preload_libraries = '$libdir/sql-profiler'
```

2. Restart the Postgres server.
3. Using the Query tool or the psql command line interface, run the `sql-profiler.sql` script in the maintenance database on the server you want to profile. If you are using:
  - PostgreSQL, the default maintenance database is `postgres`.
  - EDB Postgres Advanced Server, the default maintenance database is `edb`.

To use the PEM Query tool to run the script:

1. Select the maintenance database in the Browser tree and select **Tools > Query tool**.
2. When the Query tool opens, select **File > Open** and select the `sql-profiler.sql` script. By default, the `sql-profiler.sql` script is located in the `contrib` folder under your Postgres installation.
3. When the script opens in the SQL Editor panel of the Query tool, select the content of the script in the SQL Editor and select **Query > Execute** to invoke the script and configure SQL Profiler.

You can also use the psql command line to invoke the configuration script. The following command uses psql to invoke the `sql-profiler.sql` script on an EDB Postgres Advanced Server database on a Linux system:

```
$ /usr/edb/as<x>/bin/psql -U postgres postgres < /usr/edb/as<x>/share/contrib/sql-profiler.sql
```

Where `<x>` is the version of the EDB Postgres Advanced Server.

After configuring SQL Profiler, it's ready to use with all databases that reside on the server.

To access SQL Profiler functionality, select the monitored Server/database in the PEM Browser tree. Select **Tools > Server > SQL Profiler**. Then select:

- **Create trace** to define a new trace.
- **Open trace** to open an existing trace.
- **Delete trace(s)** to delete one or more traces.
- **View scheduled trace(s)** to review a list of scheduled traces.

## 28.1.2 SQL profiling and analysis

Most RDBMS experts agree that inefficient SQL code is the leading cause of most database performance problems. The challenge for DBAs and developers is to locate that SQL code in large and complex systems and then optimize that code for better performance.

The SQL Profiler component allows a database superuser to locate and optimize inefficient SQL code. Microsoft's SQL Server Profiler is very similar to PEM's SQL Profiler in operation and capabilities. SQL Profiler is installed with each EDB Postgres Advanced Server instance. If you're using PostgreSQL, you must download the SQL Profiler installer and install the SQL Profiler product into each managed database instance you want to profile.

For each database monitored by SQL Profiler:

1. Edit the `postgresql.conf` file. You must include the SQL Profiler library in the `shared_preload_libraries` configuration parameter.

For Linux installations, the parameter value must include:

```
$libdir/sql-profiler
```

On Windows, the parameter value must include:

```
$libdir/sql-profiler.dll
```

2. Create the functions used by SQL Profiler in your database. The SQL Profiler installation program places a SQL script (named `sql-profiler.sql`) in the `share/postgresql/contrib` subdirectory of the main PostgreSQL installation directory on Linux systems. On Windows systems, this script is located in the `share` subdirectory. You must invoke this script on the maintenance database specified when registering the server with PEM.
3. Stop and restart the server to make the changes to take effect.

### Note

If you connected to the PEM server with the PEM client before configuring SQL Profiler, you must disconnect and reconnect with the server to enable SQL Profiler functionality. For detailed information about installing and configuring the SQL Profiler plugin, see [Installing SQL Profiler](#).

## Creating a new SQL trace

SQL Profiler captures and displays a specific SQL workload for analysis in a SQL trace. You can start and review captured SQL traces immediately or save captured traces for review later. You can use SQL Profiler to create and store up to 15 named traces.

### Creating a trace

You can use the Create Trace dialog box to define a SQL trace for any database on which SQL Profiler was installed and configured. To open the dialog box, select the database in the PEM client tree and select **Tools > Server > SQL Profiler > Create trace**.

Use the **Trace options** tab to specify details about the new trace:

- Provide a name for the trace in the **Name** field.
- Use the **User filter** field to specify the roles whose queries to include in the trace. Select **Select All** to include queries from all roles.
- Use the **Database filter** field to specify the databases to trace. Select **Select All** to include queries against all databases.
- Specify a trace size in the **Maximum Trace File Size** field. SQL Profiler terminates the trace when it reaches approximately the size specified.
- Select **Yes** in the **Run Now** field to start the trace when you select **Create**. Select **No** to enable fields on the **Schedule** tab.

Use the **Schedule** tab to specify scheduling details for the new trace:

- Use the **Start time** field to specify the starting time for the trace.
- Use the **End time** field to specify the ending time for the trace.
- Select **Yes** in the **Repeat?** field to repeat the trace every day at the times specified. Select **No** to enable fields on the **Periodic job options** tab.

Use the **Periodic job options** tab to specify scheduling details about a recurring trace. Use the **Days** section to specify the days when the job executes:

- Use the **Week days** field to select the days of the week for the trace to execute.
- Use the **Month days** field to select the days of the month for the trace to execute.
- Use the **Months** field to select the months when the trace executes.

Use the **Times** section to specify a schedule for the trace execution. Use the **Hours** and **Minutes** fields to specify the time when the trace executes.

After you complete the Create Trace dialog box, select **Create** to start the trace or to schedule the trace.

If you execute the trace immediately, the trace results appear in the PEM client.

## Opening an existing trace

To view a previous trace, select the profiled database in the PEM client tree and select **Management > SQL Profiler > Open trace**. The Open Trace dialog box opens.

Select an entry in the trace list and select **Open** to open the selected trace in the **SQL Profiler** tab.

## Filtering a trace

A filter is a named set of one or more rules, each of which can hide events from the trace view. When you apply a filter to a trace, the hidden events aren't removed from the trace but are excluded from the display.

Select **Filter** to open the Trace Filter dialog box and create a rule or set of rules that define a filter. Each rule screens the events in the current trace based on the identity of the role that invoked the event or the query type invoked during the event.

To open an existing filter, select **Open**. To define a new filter, select **Add (+)** to add a row to the table displayed on the **General** tab and provide rule details:

- Use the **Type** list to specify the trace field that the filter rule applies to.
- Use the **Condition** list to specify the type of operator for SQL Profiler to apply to the value when it filters the trace:
  - Select **Matches** to filter events that contain the specified value.
  - Select **Does not match** to filter events that don't contain the specified value.
  - Select **Is equal to** to filter events that contain an exact match to the string specified in the **Value** field.
  - Select **Is not equal to** to filter events that don't contain an exact match to the string specified in the **Value** field.
  - Select **Starts with** to filter events that begin with the string specified in the **Value** field.
  - Select **Does not start with** to filter events that don't begin with the string specified in the **Value** field.
  - Select **Less than** to filter events that have a numeric value less than the number specified in the **Value** field.
  - Select **Greater than** to filter events that have a numeric value greater than the number specified in the **Value** field.
  - Select **Less than or equal to** to filter events that have a numeric value less than or equal to the number specified in the **Value** field.
  - Select **Greater than or equal to** to filter events that have a numeric value greater than or equal to the number specified in the **Value** field.

- Use the **Value** field to specify the string, number, or regular expression to search for.

After you finish defining a rule, select **Add (+)** to add another rule to the filter. To delete a rule from a filter, select the rule and select **Delete**.

Select **Save** to save the filter definition to a file without applying the filter. To apply the filter, select **OK**.

### Deleting a trace

To delete a trace:

1. Select the profiled database in the PEM client tree.
2. Select **Management > SQL Profiler > Delete trace(s)**.
3. In the Delete Traces dialog box, select the icon to the left of a trace name to mark one or more traces for deletion.
4. Select **Delete**. The PEM client acknowledges that the selected traces were deleted.

### Viewing scheduled traces

To view a list of scheduled traces, select the profiled database in the PEM client tree. Select **Management > SQL Profiler > Scheduled traces**.

The Scheduled Traces dialog box displays a list of the traces that are awaiting execution. Select **Edit** to the left of a trace name to see detailed information about the trace:

- The **Status** field lists the status of the current trace.
- The **Enabled?** switch displays Yes if the trace is enabled, No if it is disabled.
- The **Name** field displays the name of the trace.
- The **Agent** field displays the name of the agent responsible for executing the trace.
- The **Last run** field displays the date and time of the last execution of the trace.
- The **Next run** field displays the date and time of the next scheduled trace.
- The **Created** field displays the date and time that the trace was defined.

## 28.1.3 Uninstalling SQL Profiler

The process of uninstalling SQL Profiler is platform specific.

### Uninstalling SQL Profiler from Windows hosts

If you're using SQL Profiler on a Windows host, Windows locks any files that were executed or loaded into memory. To release any locked files, you must stop the Postgres server before performing an uninstall.

To stop the service:

1. In the Control Panel select **System and Security > Administrative Tools**.
2. Double-click **Services** icon.
3. In the Services dialog box, select the service name in the list, and use the option provided on the dialog box to stop the service.

After stopping the Postgres server, delete the existing SQL Profiler query set on each node by invoking the `uninstall-sql-profiler.sql` script. By default, the script resides in the `share\contrib` directory under your EDB Postgres Advanced Server or PostgreSQL installation.

## Uninstalling SQL Profiler from RHEL hosts

To uninstall a SQL Profiler installation that resides on a Linux host, delete the existing SQL Profiler query set on each node by invoking the `uninstall-sql-profiler.sql` script.

By default, if you're using EDB Postgres Advanced Server on a Linux host, the script resides in the `share/contrib` directory under the EDB Postgres Advanced Server installation. If you're using a PostgreSQL installation on a Linux host, the script resides in the `share/contrib` directory under the PostgreSQL installation.

## 28.2 Using the Index Advisor

Index Advisor is distributed with EDB Postgres Advanced Server version 9.0 and later. Index Advisor works with SQL Profiler by examining collected SQL statements and making indexing recommendations for any underlying tables to improve SQL response time. Index Advisor works on all DML (INSERT, UPDATE, DELETE) and SELECT statements that are invoked by a superuser.

Diagnostic output from the Index Advisor includes:

- Forecasted performance benefits from any recommended indexes
- The predicted size of any recommended indexes
- DDL statements you can use to create the recommended indexes

Before using Index Advisor, you must:

1. Modify the `postgresql.conf` file on each EDB Postgres Advanced Server host, adding the `index_advisor` library to the `shared_preload_libraries` parameter.
2. Install the `Index Advisor contrib` module. To install the module, use the psql client or PEM Query tool to connect to the database, and invoke the following command:

```
\i <complete_path>/share/contrib/index_advisor.sql
```

3. Restart the server to make your changes to take effect.

Index Advisor can make indexing recommendations based on trace data captured by SQL Profiler. To open Index Advisor, select one or more queries in the SQL Profiler Trace Data pane and select **Index Advisor** from the toolbar. For more information about configuring and using Index Advisor, see [EDB Postgres Advanced Server](#).

### Note

Index Advisor can't analyze statements invoked by a non-superuser. If you attempt to analyze statements invoked by a non-superuser, the server log includes the following error:

```
ERROR: access to library "index_advisor" is not allowed
```

### Note

We recommend that you disable Index Advisor while using the `pg_dump` functionality.

## 29 Viewing system reports

You can generate the System Configuration report and Core Usage report for all locally and remotely managed servers. To generate this report, select

## Management > Reports.

Reports has following options:

- System Configuration Report (JSON)
- System Configuration Report (HTML)
- Core Usage Report (JSON)
- Core Usage Report (HTML)

Only superusers or the users with the pem\_admin role can download the System Configuration or Core Usage reports.

Information in these reports shows the latest probe run time.

## System Configuration Report

The System Configuration Report provides detailed information about the PEM Agents group, PEM Server Directory group, and custom groups listed under the browser tree. These groups can contain Postgres Enterprise Manager, PEM agent, and database servers. You can download this report in HTML and JSON formats.

The Postgres Enterprise Manager Summary provides details about:

- The Postgres Enterprise Manager backend database server version
- Application version
- User name accessing the application
- Python version
- Flask version
- Platform specific information

The summary provides information about the number of agents and servers. The Group: PEM Agents panel provides details about the PEM agent, CPU cores, disk utilization, and memory information.

The Group: PEM Server Directory panel provides details about:

- Database server version
- Host
- Port
- Database name
- Database size
- Tablespace size

The group server name depends on the group name to which the server is added.

## Core Usage report

The Core Usage report provides detailed information about the number of cores specific to:

- Server type
- Database version
- Platform and group name

The report also gives detailed information about locally managed servers:

- Type
- Host

- Port
- Platform
- Cores
- RAM

## 30 PEM command line interface

### New Feature

PEM command line interface (CLI) support is available in PEM version 8.5.0 and later for Linux platforms. You can also access PEM version 8.2 to 8.4 servers using the PEM CLI.

The PEM CLI allows you to export and import the alert templates or probes from one PEM server to another.

You can install the PEM CLI using the package `edb-pem-cli-8.5.0-1-<platform_name>` separately from the `edb.repo` repository. It creates the binary file `/usr/edb/pem/cli/bin/pem` and the license file `/usr/edb/pem/edb-pem-cli-license.txt`.

After installing the PEM CLI package, use the `./pem -h` command to see all the available options.

```
$ ./pem -h
```

```
PEM CLI (Postgres Enterprise Manager Command Line Interface)
```

```
Usage: ./pem [OPTIONS] [SUBCOMMAND]
```

```
Options:
```

```
-h,--help          Print this help message and exit
-v,--version       Show the app version and exit
```

```
Logging:
```

```
--log-file [/home/asheshvashi/.pem-cli.log]
                        Set the log file
```

```
NOTE: Directory must exist
```

```
-l,--log-level :value in {FATAL->1,ERROR->2,WARNING->3,DEBUG->5,INFO->4,VERBOSE->6} OR {1,2,3,5,4,6}
[INFO]
                        Set the log level
```

```
PEM Information:
```

```
-p,--pem-url REQUIRED (Env:PEM_SERVER_URL)
                        Set the URL for accessing Postgres Enterprise Manager
                        e.g.
                        https://127.0.0.1:8443/pem
                        https://pem_host:8443/pem
```

```
-u,--pem-user REQUIRED (Env:PEM_API_USER)
                        User to access the PEM REST API
```

```
-f,--password-file (Env:PEM_PASSWORD_FILE)
                        Location of the file, which contains the password.
                        The permission on a password file must disallow any access to world or group.
```

```
NOTE:
```

```
* The first non-empty line in the file will be treated as a
```



password.

\* Environment variable 'PEM\_API\_PASSWORD' can be used for providing the password instead of the password file

`-i,--insecure` Set the insecure connection

#### Subcommands:

|                                     |   |
|-------------------------------------|---|
| <code>import-alert-templates</code> | Import the alert template(s) into the PEM server.                     |
| <code>list-alert-templates</code>   | List the alert templates in the PEM server.                           |
| <code>export-alert-templates</code> | Export the custom (user-defined) alert templates from the PEM server. |
| <code>import-probes</code>          | Import the probe(s) into the PEM server.                              |
| <code>export-probes</code>          | Export the custom (user-defined) probes from the PEM server.          |
| <code>list-probes</code>            | List the probes in the PEM server.                                    |

You can see all the available options with the `import-alert-templates` subcommand:

```
$ ./pem import-alert-templates -h
```

Import the alert template(s) into the PEM server.

Usage: `./pem import-alert-templates [OPTIONS]`

#### Options:

|                                       |  |
|---------------------------------------|--|
| <code>-h,--help</code>                | Print this help message and exit                     |
| <code>-i,--in-file</code>             | Alert template file                                  |
| <code>--ignore-existing</code>        | Ignore the existing alert template in the PEM server |
| <code>--ignore-existing-probes</code> | Ignore the existing probes in the PEM server         |

You can see all the available options with the `list-alert-templates` subcommand:

```
$ ./pem list-alert-templates -h
```

List the alert templates in the PEM server.

Usage: `./pem list-alert-templates [OPTIONS]`

#### Options:

|  |   |
|--|---|
| <code>-h,--help</code>                           | Print this help message and exit                    |
| <code>-c,--alert-template-category :value</code> | in {ALL->2,CUSTOM->1,SYSTEM->0} OR {2,1,0} [CUSTOM] |
|  | List down the alert templates of this category      |

You can see all the available options with the `export-alert-templates` subcommand:

```
$ ./pem export-alert-templates -h
```

Export the custom (user-defined) alert templates from the PEM server

Usage: `./pem export-alert-templates [OPTIONS]`

#### Options:

|  |  |
|--|--|
| <code>-h,--help</code>                                 | Print this help message and exit                         |
| <code>-i,--id INT ...</code>                           | List of id for the custom alert templates to be exported |
| <code>-o,--out-file [Standard output (console)]</code> | Output file name   |
| <code>-w,--overwrite</code>                            | Overwrite the existing output file                       |

You can see all the available options with the `import-probes` subcommand:

```
$ ./pem import-probes -h
```

Import the probe(s) into the PEM server.

Usage: ./pem import-probes [OPTIONS]

Options:

|                   |   |
|-------------------|---|
| -h,--help         | Print this help message and exit                    |
| -i,--in-file      | Probes file   |
| --ignore-existing | Ignore the existing custom probes in the PEM server |

You can see all the available options with the `export-probes` subcommand:

```
$ ./pem export-probes -h
```

Export the custom (user-defined) probes from the PEM server

Usage: ./pem export-probes [OPTIONS]

Options:

|   |   |
|---|---|
| -h,--help                                 | Print this help message and exit            |
| -i,--probe-id INT ...                     | List of the custom probe-ids to be exported |
| -o,--out-file [Standard output (console)] | Output file name                            |
| -w,--overwrite                            | Overwrite the existing output file          |

You can see all the available options with the `list-probes` subcommand:

```
$ ./pem list-probes -h
```

List the probes in the PEM server.

Usage: ./pem list-probes [OPTIONS]

Options:

|  |                                       |
|--|---------------------------------------|
| -h,--help  | Print this help message and exit      |
| -c,--probe-category :value in {ALL->2,CUSTOM->1,SYSTEM->0} OR {2,1,0} [CUSTOM] | List down the probes of this category |

## Example

To list all the probes (system as well as custom probes) use the `-c all` option:

```
$ PEM_PASSWORD_FILE=/tmp/pass PEM_API_USER=postgres PEM_SERVER_URL=https://172.17.0.4/pem ./pem -i list-probes -c all
```

Where:

- `PEM_PASSWORD_FILE=/tmp/pass` is the password file with the password of the postgres user.
- `PEM_API_USER=postgres` is the username through which you're running the PEM CLI.
- `PEM_SERVER_URL=https://172.17.0.4/pem` is the URL of the PEM server containing the probes.
- `-i list-probes` is the `insecure` option with the `list-probes` subcommand to allow insecure connection and to list the probes.
- `-c all` is the `-c` option with the `all` value to list all the system as well as custom probes.

31 Using the Query tool

PEM contains a feature-rich interactive development environment (IDE) that allows you to issue ad hoc SQL queries against Postgres servers.

You can access the Query tool by selecting **Tools > Query tool** or through the context menu of select nodes of the Browser tree. The Query tool allows you to:

- Issue ad hoc SQL queries.
- Execute arbitrary SQL commands.
- Edit the result set of a SELECT query if it's [updatable](#).
- Configure and display current connection and transaction status.
- Save the data displayed in the output panel to a CSV file.
- Review the execution plan of a SQL statement in either text, graphical, or table format (similar to <https://explain.depesz.com>).
- View analytical information about a SQL statement.

The Query tool features two panels:

- The upper panel displays the SQL editor. You can use the panel to enter, edit, or execute a query. It also shows the **History** tab, which you can use to view the queries that ran in the session. You can use the scratch pad to hold text snippets during editing.

To reopen a closed scratch pad or open a new one, right-click the SQL editor and select **Add Panel**. You can also open new scratch pads from other panels using the same technique.

- The lower panel displays the Data Output panel. The tabbed panel displays:
  - The result set returned by a query
  - Information about a query's execution plan
  - server messages related to the query's execution
  - Any asynchronous notifications received from the server.

The Query tool toolbar

The toolbar uses context-sensitive icons that provide shortcuts to frequently performed tasks.

| Icon              | Behavior  | Shortcut  |
|-------------------|---|---|
| Open File         | Display a previously saved query in the SQL editor.   | Accesskey + O   |
| Save              | Save a query or access the <b>Save</b> menu: <ul style="list-style-type: none"><li>- Select <b>Save</b> to save the selected content of the SQL editor panel in a file.</li><li>- Select <b>Save As</b> to open a new browser dialog box and specify a new location to save the selected content of the SQL editor panel.</li></ul>   | Accesskey + S   |
| Save Data Changes | Save the data changes (insert, update, or delete) in the Data Output Panel to the server.   | F6  |
| Find              | Search, replace, or navigate the code displayed in the SQL editor: <ul style="list-style-type: none"><li>- Select <b>Find</b> to provide a search target and search the SQL editor contents.</li><li>- Select <b>Find next</b> to locate the next occurrence of the search target.</li><li>- Select <b>Find previous</b> to move to the last occurrence of the search target.</li><li>- Select <b>Persistent find</b> to identify all occurrences of the search target in the editor.</li><li>- Select <b>Replace</b> to locate and replace (with prompting) individual occurrences of the target.</li><li>- Select <b>Replace all</b> to locate and replace all occurrences of the target in the editor.</li><li>- Select <b>Jump</b> to navigate to the next occurrence of the search target.</li></ul> | Cmd+F<br>Cmd+G<br>Cmd+Shift+G<br>Cmd+Shift+F<br>Alt+G |
| Copy              | Copy the content that's currently highlighted in the Data Output panel when in View/Edit data mode.   | Accesskey + C   |

| Icon            | Behavior  | Shortcut   |
|-----------------|---|--|
| Paste           | Paste a copied row into a new row when in View/Edit data mode.  | Accesskey + P  |
| Delete          | Mark selected rows for deletion. Select <b>Save Data Changes</b> to update the content with your deletions.   | Accesskey + D  |
| Edit            | Use options on the <b>Edit</b> menu to access text editing tools. The options operate on the text displayed in the SQL editor panel when in Query tool mode:<br>- Select <b>Indent Selection</b> to indent the currently selected text.<br>- Select <b>Unindent Selection</b> to remove indentation from the currently selected text.<br>- Select <b>Inline Comment Selection</b> to enclose any lines that contain the selection in SQL-style comment notation.<br>- Select <b>Inline Uncomment Selection</b> to remove SQL-style comment notation from the selected line.<br>- Select <b>Block Comment</b> to enclose all lines that contain the selection in C-style comment notation. This option acts as a toggle. | Tab<br>Shift+Tab<br>Cmd+/<br>Cmd+.<br>Shift+Cmd+/<br>Cmd+. |
| Filter          | Set filtering and sorting criteria for the data when in View/Edit data mode. Select the down arrow to access other filtering and sorting options:<br>-Select <b>Sort/Filter</b> to open the sorting and filtering dialog box.<br>- Select <b>Filter by Selection</b> to show only the rows containing the values in the selected cells.<br>- Select <b>Exclude by Selection</b> to show only the rows that don't contain the values in the selected cells.<br>- Select <b>Remove Sort/Filter</b> to remove any previously selected sorting or filtering options.  | Accesskey + F  |
| Limit Selector  | Set a value for the maximum number of rows in a dataset.  | Accesskey + R  |
| Stop            | Cancel the currently running query.   | Accesskey + Q  |
| Execute/Refresh | Execute or refresh the query selected in the SQL editor panel. Select the down arrow to access other execution options:<br>- Select <b>Auto-Rollback</b> to roll back a transaction in case an error occurs during the transaction.<br>- Select <b>Auto-Commit</b> to commit each transaction. Any changes made by the transaction are visible to others and durable in the event of a crash.   | F5   |
| Explain         | View an explanation plan for the current query. The result of <b>EXPLAIN</b> is displayed graphically on the <b>Explain</b> tab of the output panel and in text form on the <b>Data Output</b> tab.   | F7   |
| Explain analyze | Invoke an <b>EXPLAIN ANALYZE</b> command on the current query. From the <b>Explain Options</b> menu:<br>- Select <b>Verbose</b> to display more information about the query plan.<br>- Select <b>Costs</b> to include information on the estimated startup and total cost of each plan node as well as the estimated number of rows and the estimated width of each row.<br>- Select <b>Buffers</b> to include information on buffer usage.<br>- Select <b>Timing</b> to include information about the startup time and the amount of time spent in each node of the query.<br>- Select <b>Summary</b> to include the summary information about the query plan.   |  |
| Commit          | Commit the transaction.   | Shift+CTRL+M   |
| Rollback        | Roll back the transaction.  | Shift+CTRL+R   |
| Clear           | Use options on the <b>Clear</b> menu to erase display contents:<br>- Select <b>Clear Query Window</b> to erase the content of the SQL editor panel.<br>- Select <b>Clear History</b> to erase the content of the <b>History</b> tab.  | Accesskey + L  |
| Download as CSV | Download the result set of the current query to a comma-separated list. You can specify the CSV settings through the <b>Preferences -&gt; SQL editor -&gt; CSV output</b> dialog box.   | F8   |
| Macros          | Create, edit, or clear the macros by selecting <b>Manage Macros</b> .   |  |

## The SQL editor panel

The SQL editor panel is a workspace where you can manually provide a query, copy a query from another source, or read a query from a file. The SQL editor features syntax coloring and auto-completion.

To use auto-complete, begin typing your query. When you want the editor to suggest object names or commands that might be next in your query, press **Control+Space**. For example, type `\*SELECT \* FROM\*` (with a trailing space), and then press **Control+Space** to select from a menu of auto-complete options.

After entering a query, select **Execute/Refresh** from the toolbar. The database server receives the complete contents of the SQL editor panel to execute. To execute only a section of the code that's displayed in the SQL editor, select the text that you want the server to execute, and select **Execute/Refresh**.

The message returned by the server when a command executes is displayed on the **Messages** tab. If the command is successful, the **Messages** tab displays execution details.

The **Edit** menu helps with code formatting and commenting:

- Use auto-indent to indent text to the same depth as the previous line by pressing **Return**.
- Block indent text by selecting two or more lines and pressing **Tab**.
- Implement or remove SQL-style or toggle C-style comment notation within your code.

You can also drag certain objects from the tree to save time spent typing long object names. Text containing the object name is fully qualified with the schema name. Double quotes are added if required. For functions and procedures, the function name along with parameter names are pasted into the Query tool.

## The Data Output panel

The Data Output panel displays data and statistics generated by the most recently executed query.

### Data Output tab

The **Data Output** tab displays the result set of the query in a table format. You can:

- Select and copy from the result set.
- Use the **Execute/Refresh** options to retrieve query execution information and set query execution options.
- Select **Download as CSV** to download the content of the **Data Output** tab as a comma-delimited file.
- Edit the data in the result set of a **SELECT** query if it's updatable.

A result set is updatable if:

- All columns are either selected directly from a single table or they aren't actually a table column (for example, the concatenation of two columns). You can edit only columns that are selected directly from the table. Other columns are read-only.
- All the primary key columns or OIDs of the table are selected in the result set.

Any columns that are renamed or selected more than once are also read-only.

#### Note

To work with an updatable query result set, you must have psycopg2 driver version 2.8 or later installed.

Editable and read-only columns are identified using pencil and lock icons in the column headers.

An updatable result set is similar to the data grid in View/Edit Data mode, and you can modify it in the same way.

If auto-commit is off, data changes are made as part of the ongoing transaction. If no transaction is ongoing, a new one is started. The data changes aren't committed to the database unless the transaction is committed.

If any errors occur while saving (for example, trying to save a NULL into a column with a NOT NULL constraint), the data changes are rolled back to a savepoint to ensure any previously executed queries in the ongoing transaction aren't rolled back.

All rowsets from previous queries or commands that are displayed in the Data Output panel get discarded when you invoke another query. Open another Query tool browser tab to keep your previous results available.

## Explain tab

To generate the Explain or Explain Analyze plan of a query, select **Explain** or **Explain Analyze** in the toolbar.

You can select more options related to **Explain** and **Explain Analyze** from the menu.

### Note

PEM generates the Explain Analyze plan in JSON format.

On successful generation of an Explain plan, three tabs/panels appear under the **Explain** panel.

## Graphical tab

To download the plan as an SVG file, select **Download** in the top-left corner of the Explain canvas. **Download as SVG** isn't supported on Internet Explorer.

The query plan that accompanies **Explain Analyze** is available on the **Data Output** tab.

## Analysis tab

The **Analysis** tab shows the plan details in table format, generating a format similar to the one available at [explain.depsez.com](https://explain.depsez.com). Each row of the table represents the data for an Explain Plan node. The output can contain the node information, exclusive timing, inclusive timing, actual versus planned rows, actual rows, planned rows, or loops. Child rows of the selected row are marked with an orange dot.

If the percentage of the exclusive/inclusive timings of the total query time is:

- Greater than 90, red appears
- Greater than 50, orange appears
- Greater than 10, yellow appears

If the planner has misestimated the number of rows (actual verse planned) by:

- 10 times, yellow appears
- 100 times, orange appears
- 1000 times, red appears

## Statistics tab

The **Statistics** tab displays information in two tables:

- Statistics per Node Type tells you how many times each node type was referenced.
- Statistics per Table tells you how many times each table was referenced by the query.

## Messages tab

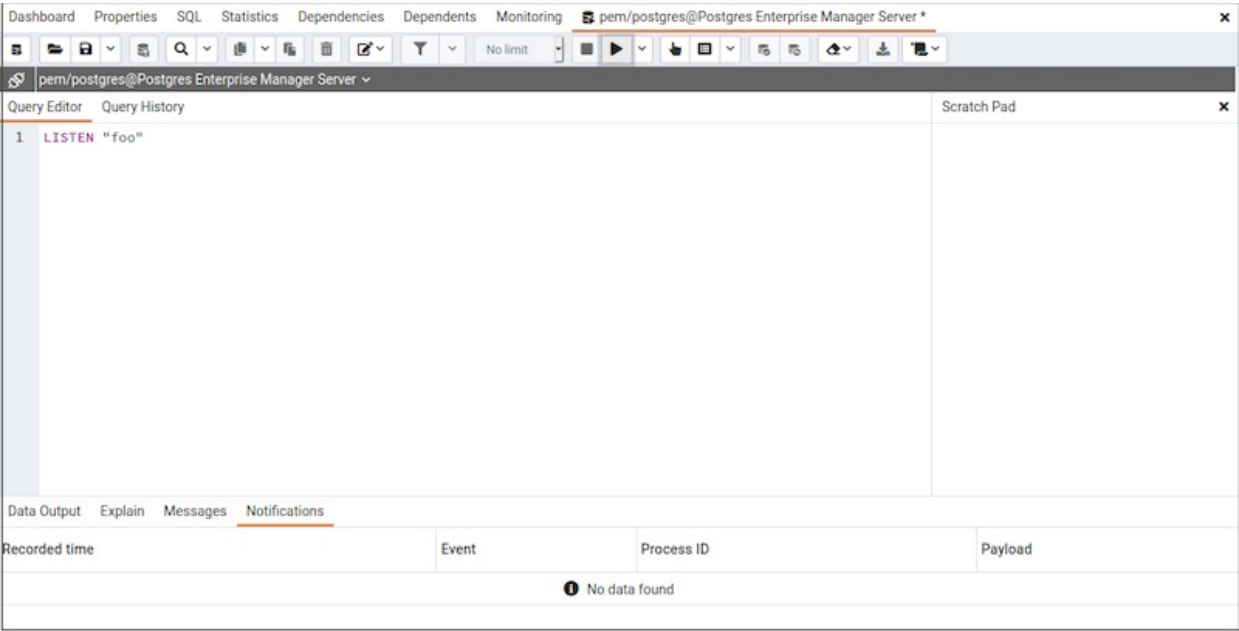
Use the **Messages** tab to view information about the most recently executed query. If the server returns an error, the error message appears on the **Messages** tab, and the syntax that caused the error is underlined in the SQL editor. If a query succeeds, the **Messages** tab shows how long the query took to complete and how many rows were retrieved.

Notifications tab

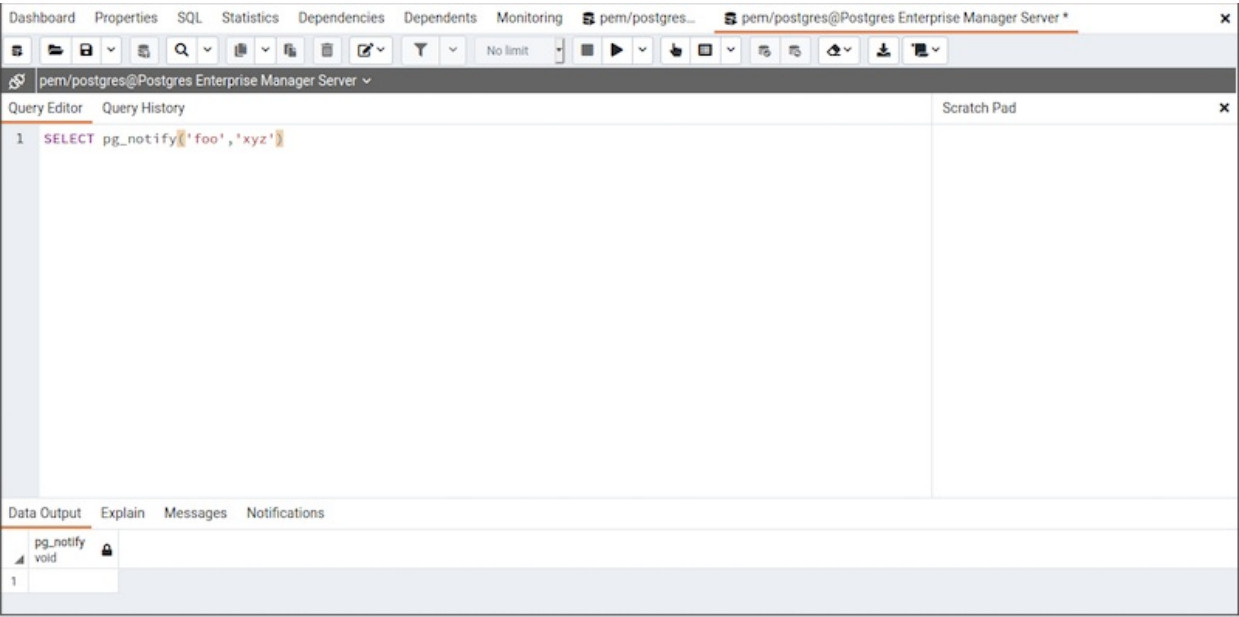
Use the **Notifications** tab to view the notifications using the PostgreSQL Listen/Notify feature. For more details, see the [PostgreSQL documentation](#).

For example:

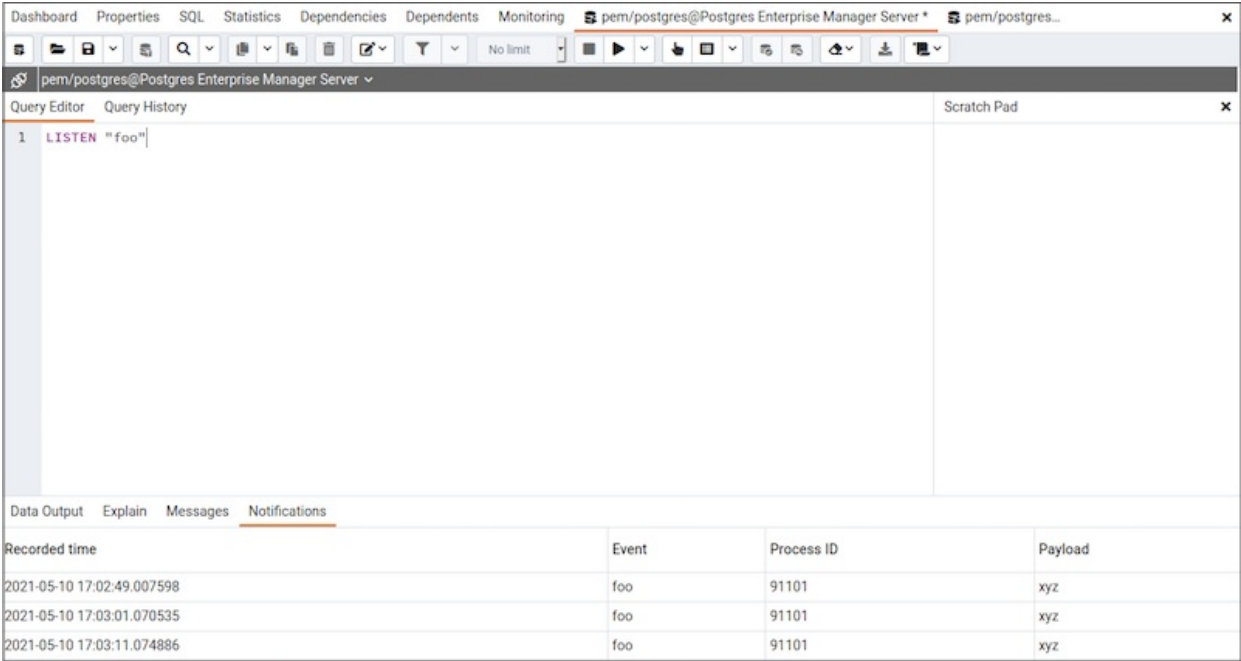
- 1. Execute `LISTEN "foo"` in first Query tool session.



- 2. In the another Query tool session, execute the `Notify` command or `pg_notify` function to send the notification of the event together with the payload.



- 3. You can observe the **Notification** tab in the first Query tool session where it shows the recorded time, event, process ID, and the payload of the channel.



Query History panel

Use the **Query History** tab to review activity for the current session. The **Query History** tab displays information about recent commands including:

- The date and time that a query was invoked.
- The text of the query.
- The number of rows returned by the query.
- The amount of time it took the server to process the query and return a result set.
- Messages returned by the server (not noted on the **Messages** tab).
- The source of the query (indicated by icons corresponding to the toolbar).

You can show or hide the queries generated internally by pgAdmin (during View/Edit Data or Save Data operations).

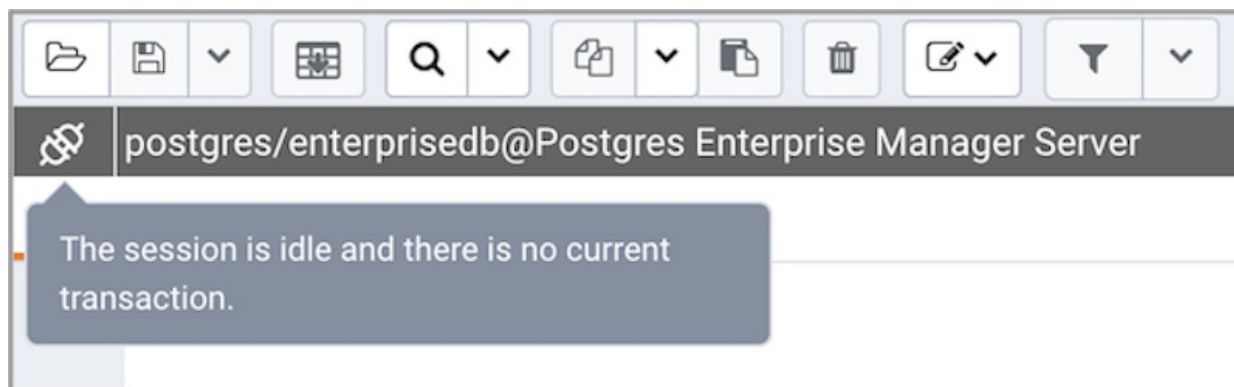
To erase the content of the **Query History** tab, select **Clear > Clear history**.

Query history is maintained across sessions for each database on a per-user basis when running in Query tool mode. In View/Edit Data mode, history isn't retained. By default, the last 20 queries are stored for each database. You can adjust this in `config_local.py` by overriding the `MAX_QUERY_HIST_STORED` value.

Connection status

Use **Connection Status** to view the current connection and transaction status by selecting the status in the Query tool.





## Macros

Query tool macros enable you to execute predefined SQL queries by pressing a single key. Predefined queries can contain the placeholder `$SELECTION$`. When the macro executes, the placeholder is replaced with the currently selected text in the Query Editor pane of the Query tool.

To create a macro:

1. In the Query tool, select **Macros > Manage Macros**.
2. Select the key you want to use. Enter the name of the macro and the query.
3. Optionally, include the selection placeholder.
4. Select **Save**.

To clear a macro, in the Manage Macros dialog box, select the macro and select **Clear**. Respond **Yes** to the prompt.

To clear all macros, select **Clear** next to **Key**. Respond **Yes** to the prompt.

To execute a macro, select the shortcut keys, or select it from the **Macros** menu.

## 32 Using the Schema Diff tool

Schema Diff compares schema objects between two database schemas. Use the **Tools** menu to access Schema Diff.

Schema Diff allows you to:

- Compare and synchronize the database schemas (from source to target).
- Visualize the differences between database schemas.
- List the differences in SQL statement for target schema objects.
- Generate synchronization scripts.

To open the selection panel, select **Tools > Schema Diff**. Select the source and target servers, databases, and schemas to compare. After selecting the objects, select **Compare**.

### Note

The source and target databases must use the same major version.

You can open multiple copies of Schema Diff in individual tabs. To specify whether to open Schema Diff in a new browser tab, select the Preferences dialog box. Set **Open in new browser tab** to **true**.

The Schema Diff panel is divided into two panels: an Object Comparison panel and a DDL Comparison panel.

## Schema Diff Object Comparison panel

In the Object Comparison panel, you can select the source and target servers of the same major version, databases, and schemas to compare. You can select any server listed under the browser tree whether it's connected or disconnected. If you select a server that isn't connected, then you must enter the password before using that server.

Select the databases to compare. The databases can be the same or different, and from the same or different servers.

Select the source and target schemas to compare.

After you select servers, databases, and schemas, select **Compare** to get the comparison results.

Use the lists of **Functions**, **Materialized Views**, **Tables**, **Trigger Functions**, **Procedures**, and **Views** to view the DDL statements of all the schema objects.

To filter the schema objects, select **Filter** in the upper-right corner of the Object Comparison panel. Filter the schema objects according to these criteria:

- **Identical** — If the object is found in both schemas with the same SQL statement, then the comparison result is identical.
- **Different** — If the object is found in both schemas with different SQL statements, then the comparison result is different.
- **Source Only** — If the object is found only in source schema and not in target schema, then the comparison result is source only.
- **Target Only** — If the object is found only in target schema and not in source schema, then the comparison result is target only.

Select any of the schema objects in the Object Comparison panel to display the DDL statements for that object in the DDL Comparison panel.

## Schema Diff DDL Comparison panel

The DDL Comparison panel displays three columns:

- The first column displays the DDL statement of the object from the source schema.
- The second column displays the DDL statement of the object from the target schema.
- The third column displays the difference in the SQL statement of the target schema object.

To check for differences in the SQL statements, review the DDL statements of all the schema objects.

The Schema Diff tool can generate a SQL script with the differences found in the target schema object. The SQL script compares the target schema object to the SQL statement of the source schema object. To generate the script:

1. In the Object Comparison panel, select the check boxes of the schema objects.
2. Select **Generate Script**.

To open the Query tool in a new tab and display the differences in the SQL statement in the Query Editor:

1. Select the schema objects.
2. Select **Generate Script**.

If you select the schema object to check the difference generated in the DDL Comparison panel but don't select the check box of the schema object, then PEM opens the Query tool in a new tab. The Query tool displays the differences in the SQL statements in the Query Editor.

To copy the difference generated in the DDL Comparison panel, select **Copy**.

Apply the SQL Statement in the target schema to synchronize the schemas.

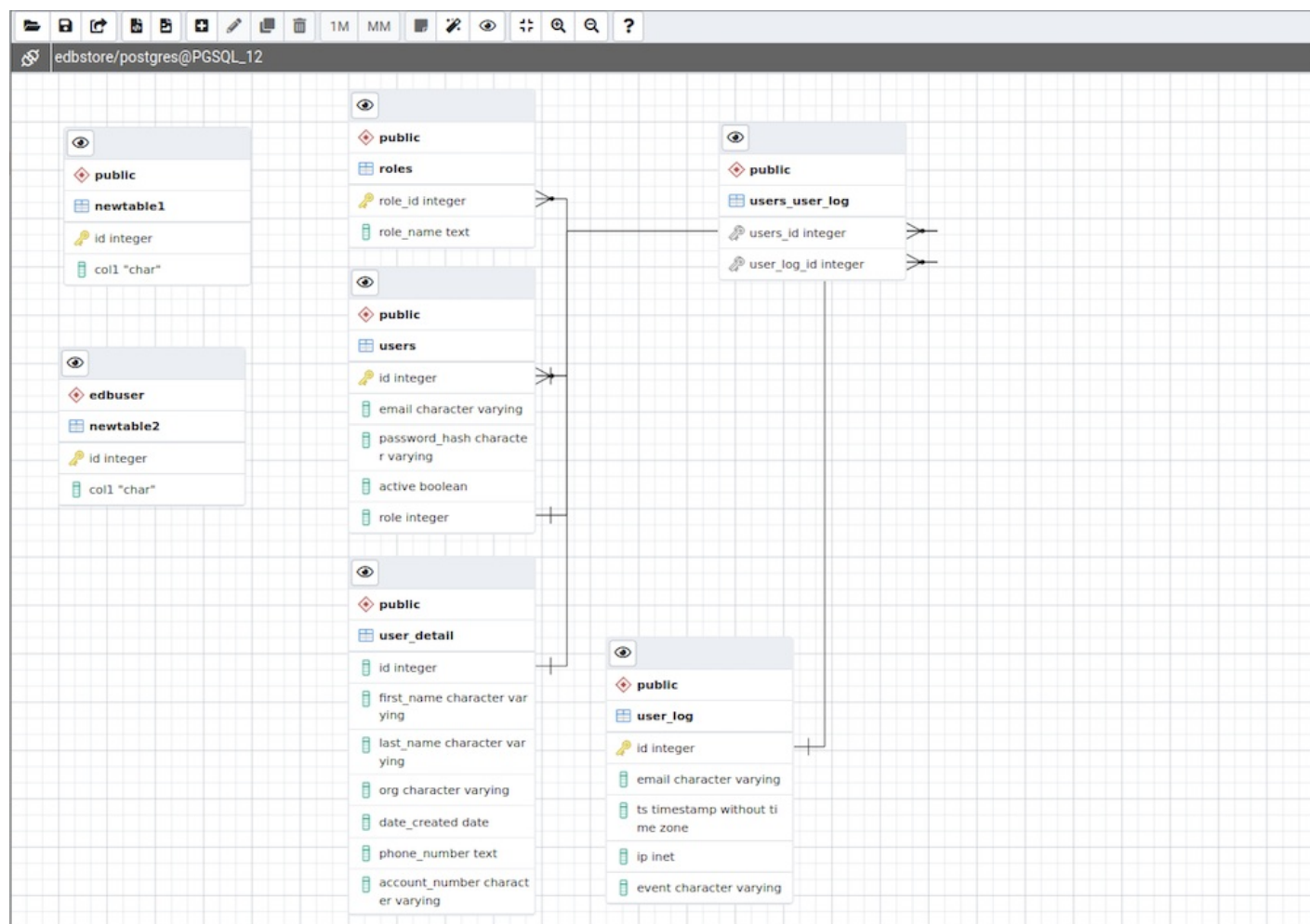
## 33 Using the ERD tool

**\*\*New feature \*\***

The ERD tool is available in PEM 8.1.0 and later.

The Entity-Relationship Diagram (ERD) tool is a database design tool that provides a graphical representation of database tables, columns, and inter-relationships. ERD can give sufficient information for the database administrator to follow when developing and maintaining the database. To access the ERD tool, select **Tools > ERD Tool**. The ERD tool allows you to:

- Design and visualize the database tables and their relationships.
- Add notes to the diagram.
- Align the tables and links for a cleaner look.
- Save the diagram and open it later to continue working on it.
- Generate ready-to-run SQL from the database design.
- Generate the database diagram for an existing database.



You can open multiple copies of the ERD tool in individual tabs simultaneously. To close a copy of the ERD tool, select the **X** in the upper-right corner of the tab bar.

## Toolbar

The ERD Tool toolbar uses context-sensitive icons that provide shortcuts to frequently performed tasks.

## File options

| Icon      | Behavior  | Shortcut         |
|-----------|---|------------------|
| Open File | Load a saved diagram.   | Ctrl + O         |
| Save      | Save changes to a saved diagram or save the diagram to a file.        | Ctrl + S         |
| Save as   | Open a new dialog box and specify a new location to save the diagram. | Ctrl + Shift + S |

## Export options

| Icon           | Behavior   | Shortcut          |
|----------------|--|-------------------|
| Generate SQL   | Generate the DDL SQL for the diagram and open a Query tool with the generated SQL ready for execution. | Option + Ctrl + S |
| Download image | Save the ERD diagram in an image format.   | Option + Ctrl + I |

## Editing options

| Icon            | Behavior   | Shortcut              |
|-----------------|--|-----------------------|
| Add table       | Add a new table to the diagram. In the table dialog box, enter the table details.                  | Option/Alt + Ctrl + A |
| Edit table      | Edit a selected table on the diagram. In the table dialog box, enter the table details.            | Option/Alt + Ctrl + E |
| Clone table     | Clone the complete table structure, name it with a auto-generated name, and put it in the diagram. | Option/Alt + Ctrl + C |
| Drop table/link | Drop a selected table or link.   | Option/Alt + Ctrl + D |

## Table relationship options

| Icon | Behavior   | Shortcut              |
|------|--|-----------------------|
| 1M   | Open a one-to-many relationship dialog box to add a relationship between two tables. The selected table is the referencing table and has the <b>many</b> endpoint of the link. | Option/Alt + Ctrl + O |
| MM   | Open a many-to-many relationship dialog box to add a relationship between two tables. This option creates a linked table from the selected columns of the two relating tables. | Option/Alt + Ctrl + M |

## Utility Options

| Icon          | Behavior   | Shortcut               |
|---------------|--|------------------------|
| Add/Edit note | Make notes on table nodes while designing the database.                            | Option/Alt + Ctrl + N  |
| Auto align    | Align all tables and links for a cleaner look.                                     | Option/Alt + Ctrl + L  |
| Show details  | Toggle the column details. You can also choose to show few or more column details. | Option/Alt + Shift + D |

## Zoom options

| Icon        | Behavior  | Shortcut               |
|-------------|---|------------------------|
| Zoom to fit | Zoom in/out automatically and fit all the tables to the view. | Option/Alt + Shift + F |

| Icon     | Behavior                 | Shortcut                 |
|----------|--------------------------|--------------------------|
| Zoom in  | Zoom in on the diagram.  | Option/Alt + Shift + "+" |
| Zoom out | Zoom out on the diagram. | Option/Alt + Shift + "-" |

Table dialog

The table dialog box allows you to:

- Change the table structure details.
- Edit an existing table or add a new one.
- See [Table dialog](#) for information on different fields.

Table node

The table node shows table details in a graphical representation:

- On the top bar:
  - To show column details, select **Details toggle**.
  - When you add a note, **Note** becomes enabled. Select **Note** to changes notes.
- The first row displays the table's schema name, like `public` in the example.
- The second row displays the table's name, like `users` in the example.
- The remaining rows display the table's column names and their data types.
  - A column that is a primary key displays as a lock key, like `id integer` in the example.
  - Other columns display as a green column, like `email character varying` in the example.
- Select the node to drag it on the canvas.
- To open the table dialog box and change the table details, either double-click the table node or select **Edit** on the toolbar. See [Table dialog](#) for information on different fields.

One-to-many link dialog box

One to many relation

General

Local Table

(public) roles

Local Column

role\_id

Referenced Table

(public) users

Referenced Column

role

Cancel

OK

The one-to-many link dialog box allows you to:

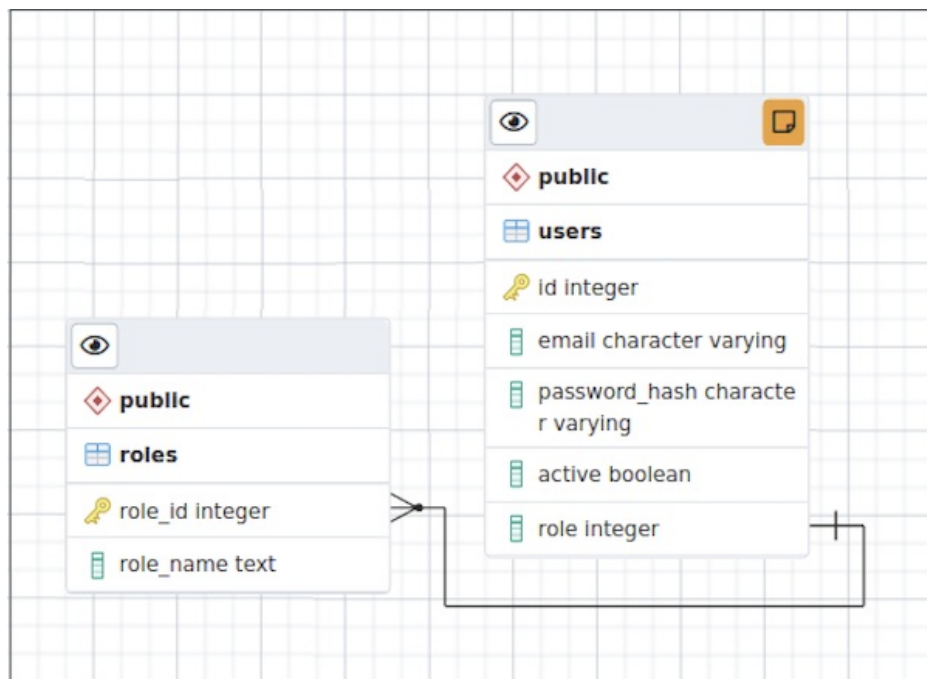
- Add a foreign key relationship between two tables.
- Select a Local Table that references another table. The Local Table has the **many** end point.
- Select a Local Column that references another column.
- Select the Referenced Table. The Referenced Table has the **one** end point.
- Select the Referenced Column.

### Many-to-many link dialog box

The many-to-many link dialog box allows you to:

- Add a many-to-many relationship between two tables.
- Create a relation table with linked columns that are derived from the two tables.
- Select the Left Table, which is the first table that is linked. The Left Table has the **one** endpoint of the link with the new relation table.
- Select the Left table Column, which is the first table's column. The Left table Column is a primary key.
- Select the Right Table, which is the second table that is linked. The Right Table has the **one** endpoint of the link with the new relation table.
- Select the Right table Column, which is the second table's column. The Right table Column is a primary key.

### Table link



The table link shows the relationship between tables:

- The single-line endpoint of the link shows the column that is the reference.
- The three-line endpoint of the link shows the column that references.
- If one of the columns that is a reference or that references is removed from the table, then the link drops.
- Select the link to drag it on the canvas.

## Table notes

To add a note to a table:

1. On the toolbar, select **Add/Edit note**.
2. In the window, enter your note.

When a note is added to a table, **Note** becomes enabled on the Table node. To check or update notes, select **Note**.

## 34 Postgres Enterprise Manager

Welcome to Postgres Enterprise Manager (PEM). Postgres Enterprise Manager (PEM) consists of components that provide the management and analytical functionality for your EDB Postgres Advanced Server or PostgreSQL database. PEM is based on the Open Source pgAdmin 4 project.

pgAdmin is the leading Open Source management tool for Postgres, the world's most advanced Open Source database. pgAdmin 4 is a comprehensive [database](#) design and management system. pgAdmin 4 is designed to meet the needs of both novice and experienced Postgres users alike, providing a powerful graphical interface that simplifies the creation, maintenance and use of database objects.

### 34.1 PEM Getting Started

You can use either a graphical installer or an RPM package to install the PEM server and PEM agent; for detailed installation instructions, see [Postgres Enterprise Manager Installation docs](#).

- [PEM Architecture](#)
- [PEM Server Logon](#)
- [Managing Configuration Settings](#)
- [Roles for managing PEM](#)

Each **server** in the PEM client tree control defines a set of connection and authentication properties that creates a connection to a Database server. The server can optionally bind the server to a PEM agent, defining a **managed** server, or not specify an agent binding, defining an **unmanaged** server. The process of defining the connection to a server is referred to as registering a server. You can use the **Auto Discovery** dialog to simplify the process of registering a managed server, or register a server manually.

You can (optionally) use the **Server Group** dialog to create server groups to organize the server connections within the tree control for easier server management. To open the **Server Group** dialog, open the **Object** menu, and select **Server Group** from the **Create** menu.

- [The Group Dialog](#)
- [Automatic Discovery of Servers](#)
- [Defining a server](#)
- [Defining and Monitoring Postgres instances on AWS](#)
- [Connect to server](#)
- [Controlling a Server](#)
- [Connection error](#)
- [Enable Kerberos Authentication](#)

### 34.1.1 PEM Architecture

Postgres Enterprise Manager (PEM) is a tool designed to monitor and manage multiple Postgres servers through a single GUI interface. PEM is capable of monitoring the following areas of the infrastructure:

Note: The term Postgres refers to either PostgreSQL or EDB Postgres Advanced Server.

- **Hosts** - One or more servers (physical or virtual) and their operating systems.
- **Servers** - One or more instances of PostgreSQL or EDB Postgres Advanced Server or EDB Postgres Extended (formerly known as 2ndQPostgres) running on a host.
- **Databases** - One or more databases and the schema objects (tables, indexes, etc.) within them.

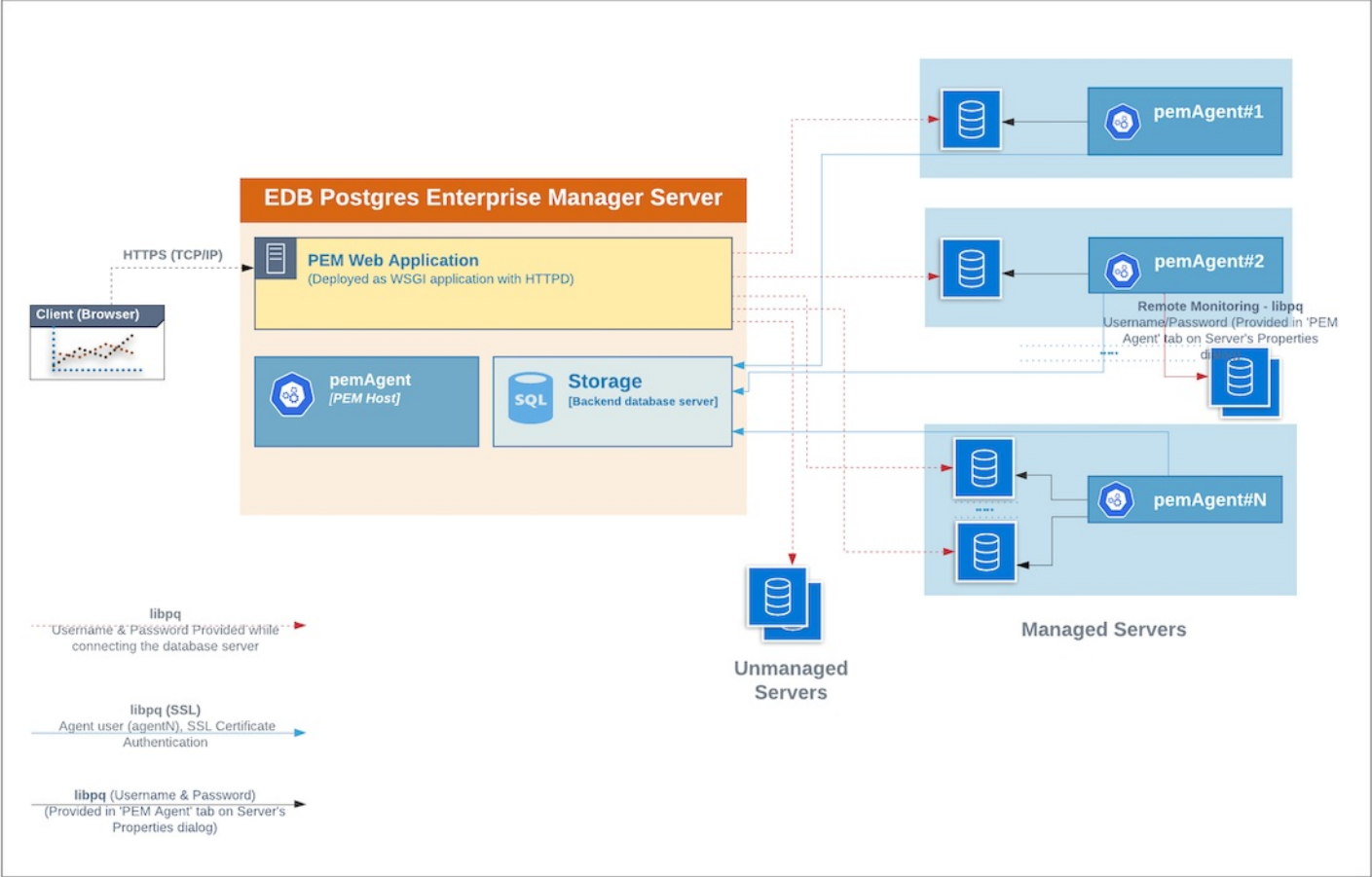
PEM consists of a number of individual software components; the individual components are described below.

- **PEM Server** - The PEM Server is used as the data repository for monitoring data and as a server to which both Agents and Clients connect. The PEM server consists of an instance of PostgreSQL and an associated database for storage of monitoring data, and a server that provides web services.
- **PEM Agent** - The PEM Agent is responsible for executing tasks and reporting statistics from the Agent host and monitored Postgres instances to the PEM server. A single PEM Agent can monitor multiple installed instances of Postgres that reside on one or many hosts.
- **PEM Web Client** - The PEM web interface allows you to manage and monitor Postgres servers and utilize PEM extended functionality. The web interface software is installed with the PEM server and is accessed via any supported web browser.
- **SQL Profiler** - SQL Profiler is a Postgres server plugin to record the monitoring data and query plans to be analysed by the SQL Profiler tool in PEM. This is an optional component of PEM, but the plugin must be installed into each instance of Postgres with which you wish to use the SQL Profiler tool. The SQL Profiler may be used with any supported version of an EnterpriseDB distribution of a PostgreSQL server or Advanced Server (not just those managed through the PEM server). See the [PEM SQL Profiler configuration docs](#) for details and supported versions.

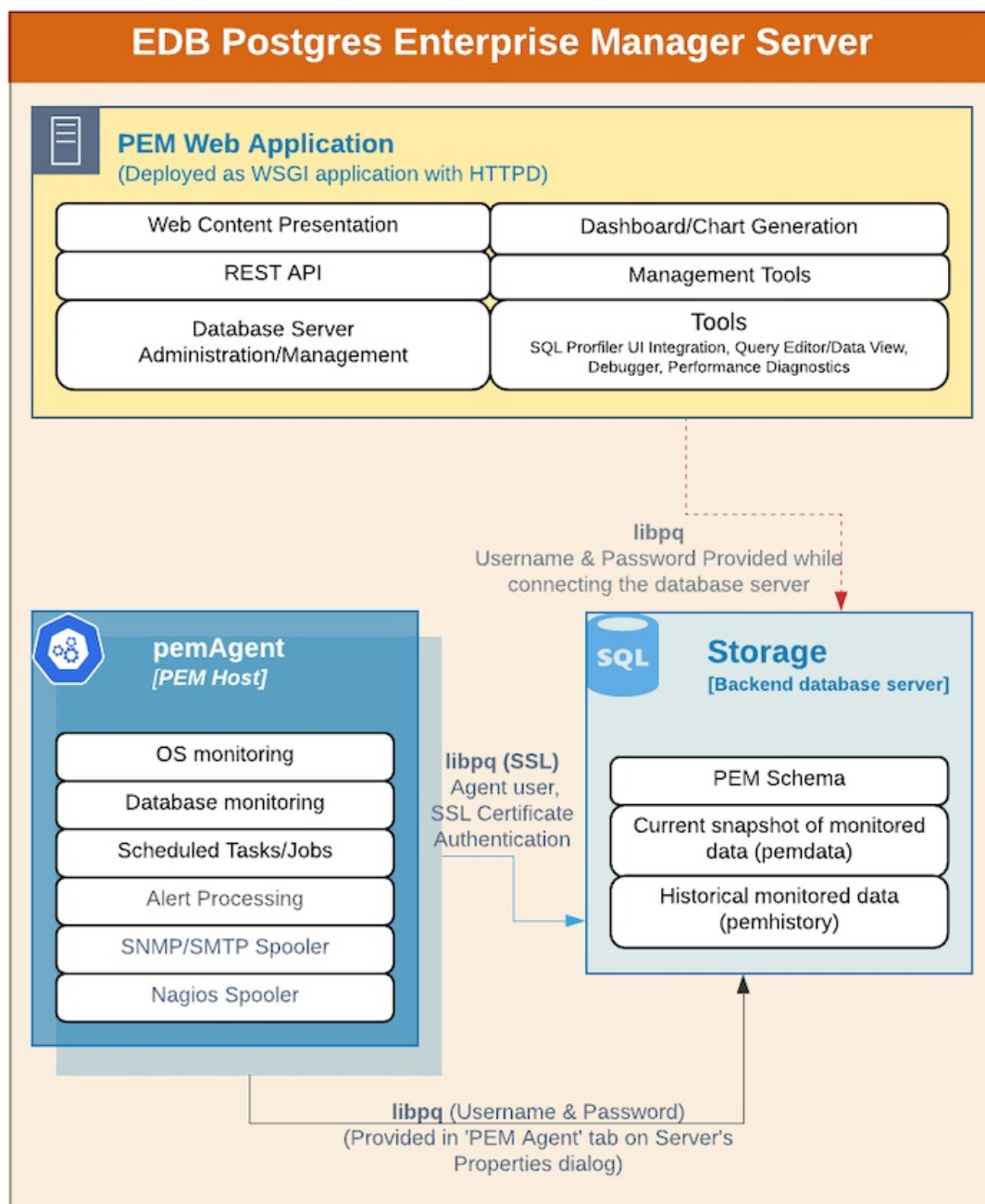
#### PEM architecture

The following architectural diagram illustrates the relationships between the PEM server, clients, and managed as well as unmanaged Postgres servers.





The PEM Server



The PEM server consists of an instance of Postgres, an instance of the Apache web-server providing web services to the client, and a PEM Agent. PEM utilizes a server-side cryptographic plugin to generate authentication certificates.

The instance of Postgres (a database server) and an instance of the Apache web-server ( HTTPD) can be on the same host or on separate hosts.

- **Postgres Instance (Database server)** - This is the backend database server. It hosts a database named **pem** which acts as the repository for PEM Server. The **pem** database contains several schemas that store metric data collected from each monitored host, server, and database.

#### Note

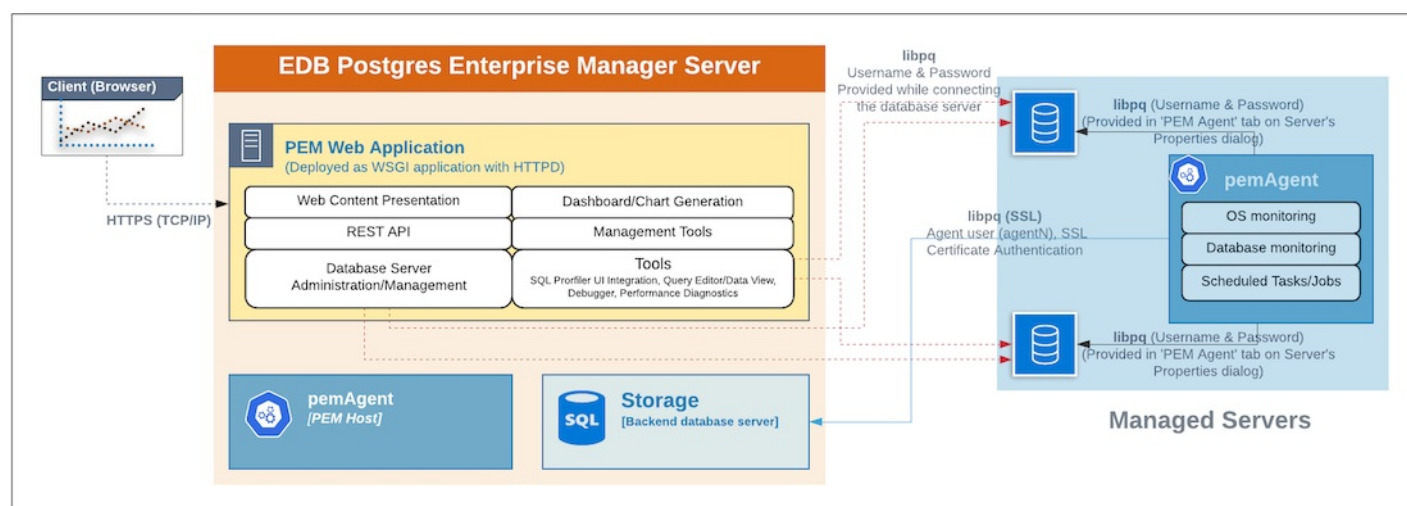
All the PEM features are available irrespective of which backend database server you select, PostgreSQL or EDB Postgres Advanced Server.

- **pem** - This schema is the core of the PEM application. It contains the definitions of configuration functions, tables, or views required by the application.
  - **pemdata** - This schema stores the current snapshot of the monitored data.
  - **pemhistory** - This schema stores the historical monitored data.
- **Apache Web Server (HTTPD)** - The PEM Web Application is deployed as a WSGI application with HTTPD to provide web services to the client. It is comprised of the following:

- **Web content presentation** - The presentation layer is created by the Web Application (for example Browser, login page,...).
- **Rest API** - The REST API allows integration with other apps and services.
- **Database Server Administration/Management** - Database server administration and management activities like CREATE, ALTER, DROP, etc. can be performed for managed as well as unmanaged servers.
- **Dashboard/Chart generation** - Internally, the web application includes functionality that generates Dashboards and Charts.
- **Management Tools** - The Audit Manager, Capacity Manager, Log Manager, Postgres Expert, Postgres Log Analysis Expert, and the Tuning Wizard are made available in the Web Application.
- Other tools provide functionality on managed or unmanaged servers:
  - **SQL Profiler UI Integration** - SQL Profiler generates easily analyzed traces of session content.
  - **Query Editor/Data View** - The Query editor allows you to query, edit, and view data.
  - **Debugger** - The Debugger helps you debug queries.
  - **Performance Diagnostics** - Performance Diagnostics help you analyze the performance of Advanced Server.

We recommended that you use a dedicated machine to host production instances of the PEM backend database. The host may be subject to high levels of data throughput, depending on the number of database servers that are being monitored and the workloads the servers are processing.

## The PEM Agent



The PEM Agent is responsible for the collection of monitoring data from the machine and operating system, as well as from each of the Postgres instances to which they are bound. Each PEM Agent can monitor one physical or virtual machine and is capable of monitoring multiple database servers locally - installed on the same system, or remotely - installed on other systems. It is also responsible for executing other tasks that may be scheduled by the user (for example, server shutdowns, SQL Profiler traces, user-defined jobs).

A PEM Agent is installed by default on the PEM Server along with the installation of the PEM Server. It is generally referred to as a PEM Agent on the PEM Host. Separately, the PEM Agent can also be installed on the other servers hosting the Postgres instances to be monitored using PEM.

Whether monitoring locally or remotely, the PEM Agent connects to the PEM Server using PostgreSQL's libpq, using SSL certificate-based authentication. The PEM Agent installer in Windows and pemworker CLI in Linux is responsible for registering each agent with the PEM Server, and generating and installing the required certificates.

Please note that there is only one-way traffic between the PEM Agent and PEM Server; the PEM Agent always connects to the PEM Server.

The PEM Agent must be able to connect to each database server that it monitors. This connection is made over a TCP/IP connection (or optionally a Unix Domain Socket on Unix hosts), and may optionally use SSL. The user must configure the connection and authentication to the monitored server.

Once configured, each agent collects statistics and other information on the host and each database server and database that it monitors. Each piece of information is known as a **metric** and is collected by a **probe**. Most probes will collect multiple metrics at once for efficiency. Examples of the metrics collected include:

- Disk I/O statistics
- Network statistics
- Database server version string

- Database server configuration option (GUC) values
- Table access statistics
- Table and index sizes

For a list of PEM probes, see [Probes](#).

By default, the PEM Agent bound to the database server collects the OS/Database monitoring statistics and also runs any scheduled tasks/jobs for that particular database server, storing data in the pem database on the PEM server.

The Alert processing, SNMP/SMTP spoolers, and Nagios Spooler data is stored in the pem database on the PEM server and is then processed by the PEM Agent on the PEM Host by default. However, processing by other PEM Agents can be enabled by adjusting the SNMP/SMTP and Nagios parameters of the PEM Agents.

To see more information about these parameters see [Server Configuration](#).

## The PEM Web Client

The PEM client is a web-based application that runs in supported browsers. The client's web interface connects to the PEM server and allows direct management of managed or unmanaged servers, and the databases and schemas that reside on them.

The client allows you to use PEM functionality that makes use of the data logged on the server through features such as the dashboards, the Postgres Log Analysis Expert, and Capacity Manager.

## The SQL Profiler Plugin

You are not required to install the SQL Profiler plugin on every server, but you must install and configure the plugin on each server on which you wish to use the SQL Profiler. You may also want to install and configure SQL Profiler on un-monitored development servers. For ad-hoc use also, you may temporarily install the SQL Profiler plugin.

The plugin is installed with the EDB Postgres Advanced Server distribution but must be installed separately for use with PostgreSQL. The SQL Profiler installer is available from the [EnterpriseDB website](#).

SQL Profiler may be used on servers that are not managed through PEM, but to perform scheduled traces, a server must have the plugin installed, and must be managed by an installed and configured PEM agent.

For more information about using SQL Profiler, see the [PEM SQL Profiler Configuration docs](#)

### 34.1.2 PEM Server Logon

The PEM web interface uses Apache to connect to the PEM server on port 8080 of the IP address on which the PEM server is installed. To connect to PEM, open your browser of choice, and navigate to:

```
<ip_address_of_PEM_host>:8080/pem
```

Where `ip_address_of_PEM_host` specifies the IP address of the host of the PEM server.

If the PEM Server is using the `Kerberos Authentication`, you will be directly connected to the PEM interface without having to login.



Use the fields on the Login window to authenticate yourself with the PEM server:

- Provide the name of a `pem` database user in the `Username` field. Users logon to PEM using user credentials setup as `login roles` on the PostgreSQL database used by the PEM server. By default, the `postgres` superuser account will be used for the initial logon.

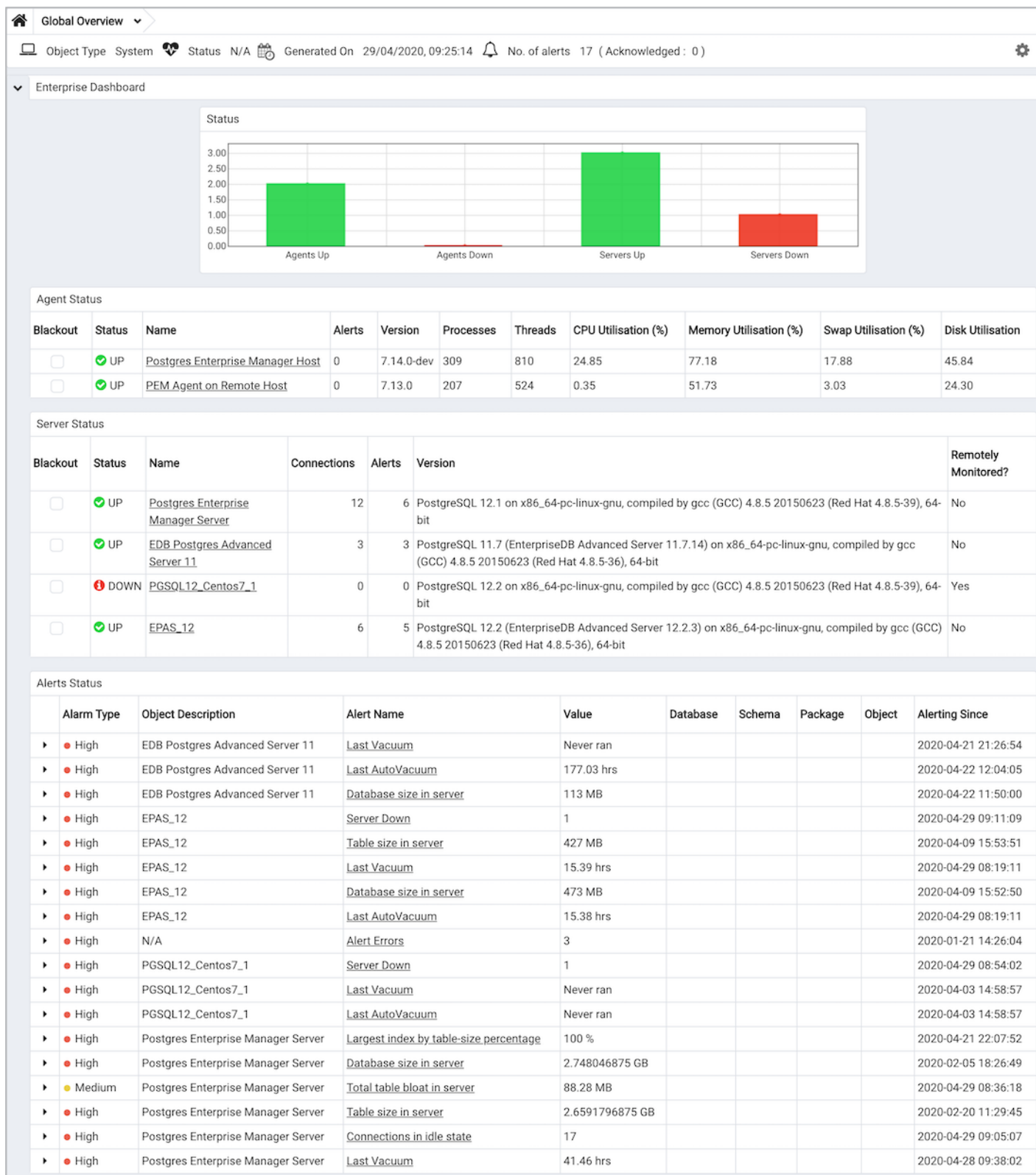
We strongly recommend you create an individual role for each user. You can create a login role with the `CREATE ROLE` SQL statement, or by defining a role with the PEM client `Create - Login/Group Role` dialog. To access the dialog, connect to the PEM server database; right-click the `Login/Group Roles` node in the tree control, and select `New Login Role...` from the `Create` pull-aside menu. Roles must be granted permissions and role memberships to properly use PEM:

- users that are members of the `pem_user` role are essentially `read-only` users; they may view dashboards, change the database server connection options, but they will not be able to install agents or configure the server directory, alerts, probes, or run any of the wizard/dialog based components of PEM.
  - users that are members of the `pem_admin` role have the same read permissions as members of the `pem_user` role, plus sufficient privileges to configure the servers, directory, alerts and probes.
  - `administrative` users must be added to the `pem_admin` role and explicitly granted the create role privilege. In addition to the permissions granted through membership in the `pem_admin` role, the create role privilege allows an administrator to create additional `pem` users, and to install and register new agents.
  - users can be member of one of the `PEM roles` to give right to run a particular component, to manage, or to configure PEM.
- Provide the password associated with the user in the `Password` field.

#### Note

From PEM 8.2 onwards, the Two-Factor Authentication support is added. For more information see [PEM Installation docs](#) on [EDB Doc Website](#).

After providing your credentials, click `Login` to connect to the PEM client. PEM opens, displaying the `Global Overview` Dashboard:



### 34.1.3 Managing Configuration Settings

There are multiple configuration files that are read at startup by Postgres Enterprise Manager. These are as follows:

- config.py** : This is the main configuration file, and should not be modified. It can be used as a reference for configuration settings, that may be overridden in one of the following files.
- config\_distro.py** : This file is read after **config.py** and is intended for packagers to change any settings that are required for their Postgres Enterprise Manager distribution. This may typically include certain paths and file locations. This file is optional, and may be created by

packagers in the same directory as `config.py` if needed.

- `config_setup.py` : This file is read after `config_distro.py` and is intended for configuration script to change any settings that are required for their database connection information. This file is optional, and may be created by configuration script in the same directory as `config.py` if needed.
- `config_local.py` : This file is read after `config_distro.py` and is intended for end users to change any default or packaging specific settings that they may wish to adjust to meet local preferences or standards. This file is optional, and may be created by users in the same directory as `config.py` if needed.

The default `config.py` file is shown below for reference:

```
# -*- coding: utf-8 -
*-

#####
#
# Postgres Enterprise
Manager
#
# Copyright (C) 2013 - 2021, EnterpriseDB Corporation. All rights
reserved.
#
# Portions of Postgres Enterprise Manager are derived from pgAdmin 4, which
is
# released under the PostgreSQL
License.
# Copyright (C) 2013 - 2020 The pgAdmin Development
Team
#
# config.py - Core application configuration
settings
#
#####

# WARNING: config.py file will be overwritten during
upgrade.

# If you want to override the value of any parameter in conf.py
file,
# you must update those entries in the config_local.py
file.

import
builtins
import logging
import os
import
sys

# We need to include the root directory in sys.path to ensure that we
can
# find everything we need when running in the standalone
runtime.
root = os.path.dirname(os.path.realpath(__file__))
if sys.path[0] != root:
    sys.path.insert(0,
root)

from pgadmin.utils import env, IS_WIN, fs_short_path

#####
# Misc stuff
#####

# Path to the online
help.
```



```

HELP_PATH = '../../../docs/en_US/_build/html/'

# Languages we support in the
UI
LANGUAGES =
{
    'en': 'English',
    # 'zh': 'Chinese
(Simplified)',
    # 'de': 'German',
    # 'fr': 'French',
    # 'ko': 'Korean',
    # 'ja':
'Japanese',
    # 'pl': 'Polish',
    # 'ru': 'Russian',
    # 'es': 'Spanish',
}

# DO NOT CHANGE UNLESS YOU KNOW WHAT YOU ARE
DOING!
# List of modules to skip when dynamically
loading
MODULE_BLACKLIST =
['test']

# DO NOT CHANGE UNLESS YOU KNOW WHAT YOU ARE
DOING!
# List of treeview browser nodes to skip when dynamically
loading
NODE_BLACKLIST =
[]

# Data directory for storage of config settings etc. This shouldn't
normally
# need to be changed - it's here as various other settings depend on
it.
if IS_WIN:
    # Use the short path on
windows
    DATA_DIR =
os.path.realpath(
    os.path.join(fs_short_path(env('APPDATA')),
'pem')
)
else:
    DATA_DIR =
os.path.realpath(os.path.expanduser('~/.pem/'))

# An optional login banner to show security warnings/disclaimers etc.
at
# login and password recovery etc. HTML may be included for basic
formatting,
# For
example:
# LOGIN_BANNER = "<h4>Authorised Users Only!</h4>"
|
#
"Unauthorised use is strictly
forbidden."
LOGIN_BANNER = ""

#####
# Log
settings
#####

```



```

# Debug mode?
DEBUG = False

# Application log level - one
of:
# CRITICAL
50
# ERROR 40
# WARNING 30
# SQL
25
# INFO 20
# DEBUG 10
# NOTSET
0
CONSOLE_LOG_LEVEL = logging.WARNING
FILE_LOG_LEVEL =
logging.WARNING

# Log
format.
CONSOLE_LOG_FORMAT = '%(asctime)s: %(levelname)s\t%(name)s:\t%(message)s'
FILE_LOG_FORMAT = '%(asctime)s: %(levelname)s\t%(name)s:\t%(message)s'

# Log file
name
LOG_FILE = os.path.join(DATA_DIR,
'pem.log')

# Log rotation
setting
# Log file will be rotated considering values for
LOG_ROTATION_SIZE
# & LOG_ROTATION_AGE. Rotated file will be named in
format
# - LOG_FILE.Y-m-d_H-M-
S
LOG_ROTATION_SIZE = 10 # In
MBs
LOG_ROTATION_AGE = 1440 # In
minutes
LOG_ROTATION_MAX_LOG_FILES = 90 # Maximum number of backups to
retain

#####
# Server
settings
#####

# This option allows the user to host the application on a
LAN
# Default hosting is on localhost
(DEFAULT_SERVER='localhost').
# To host PEM GUI application over LAN set
DEFAULT_SERVER='0.0.0.0'
# (or a specific adaptor
address.
#
# NOTE: This is NOT recommended for production use, only for
debugging
# or testing. Production installations should be run as a WSGI
application
# behind Apache
HTTPD.
DEFAULT_SERVER =
'127.0.0.1'

```

```

# The default port on which the app server will listen if not set in
the
# environment by the
runtime
DEFAULT_SERVER_PORT =
5050

# This param is used to validate ALLOWED_HOSTS for the
application
# This will be used to avoid Host Header Injection
attack
# For how to set ALLOWED_HOSTS see netaddr
library
# For more details
https://netaddr.readthedocs.io/en/latest/tutorial_03.html
# e.g. ALLOWED_HOSTS = ['192.0.2.0/28',
'::192.0.2.0/124']
# ALLOWED_HOSTS = ['225.0.0.0/8', '226.0.0.0/7',
'228.0.0.0/6']
# ALLOWED_HOSTS = ['127.0.0.1',
'192.168.0.1']
# if ALLOWED_HOSTS= [] then it will accept all ips (and application will
be
# vulnerable to Host Header Injection
attack)
ALLOWED_HOSTS = []

# Enable X-Frame-Option
protection.
# Set to one of "SAMEORIGIN", "ALLOW-FROM origin" or "" to
disable.
# Note that "DENY" is NOT supported (and will be silently
ignored).
# See https://tools.ietf.org/html/rfc7034 for more
info.
X_FRAME_OPTIONS = "SAMEORIGIN"

# The Content-Security-Policy header allows you to restrict how
resources
# such as JavaScript, CSS, or pretty much anything that the browser
loads.
# see https://content-security-policy.com/#source_list for more
info
# e.g. "default-src https: data: 'unsafe-inline' 'unsafe-
eval';"
CONTENT_SECURITY_POLICY = "default-src ws: https: data: blob: "
\
        "'unsafe-inline' 'unsafe-
eval';"

# The Strict-Transport-Security header tells the browser to convert all
HTTP
# requests to HTTPS, preventing man-in-the-middle (MITM)
attacks.
# e.g. 'max-age=31536000; includeSubDomains'
STRICT_TRANSPORT_SECURITY = "max-age=31536000; includeSubDomains"

# The X-Content-Type-Options header forces the browser to honor the
response
# content type instead of trying to detect it, which can be abused
to
# generate a cross-site scripting (XSS)
attack.
# e.g. nosniff
X_CONTENT_TYPE_OPTIONS = "nosniff"

```

```

# The browser will try to prevent reflected XSS attacks by not loading
the
# page if the request contains something that looks like JavaScript and
the
# response contains the same data. e.g. '1;
mode=block'
X_XSS_PROTECTION = "1;
mode=block"

# Hashing algorithm used for password
storage
SECURITY_PASSWORD_HASH = 'pbkdf2_sha512'

# Reverse Proxy parameters
# You must tell the middleware how many proxies set each
header
# so it knows what values to
trust.
# See
https://tinyurl.com/yyg7r9av
# for more
information.

# Number of values to trust for X-Forwarded-
For
PROXY_X_FOR_COUNT = 1

# Number of values to trust for X-Forwarded-
Proto.
PROXY_X_PROTO_COUNT =
1

# Number of values to trust for X-Forwarded-
Host.
PROXY_X_HOST_COUNT = 0

# Number of values to trust for X-Forwarded-
Port.
PROXY_X_PORT_COUNT = 1

# Number of values to trust for X-Forwarded-
Prefix.
PROXY_X_PREFIX_COUNT = 0

# NOTE: CSRF_SESSION_KEY, SECRET_KEY and SECURITY_PASSWORD_SALT are
no
# longer part of the main configuration, but are stored in
the
# configuration databases 'keys' table and are auto-
generated.
WTF_CSRF_HEADERS = ['X-EdbPEM-
CSRFToken']

# Set the cache control max age for static files in flask to 1
year
SEND_FILE_MAX_AGE_DEFAULT = 31556952

# This will be added to static urls as url parameter with value
as
# APP_VERSION_INT for cache busting on version upgrade. If the value is set
as
# None or empty string then it will not be
added.
# eg - http://localhost:5050/pgadmin.css?
intver=3.13
APP_VERSION_PARAM = 'ver'

```

```

    # Add the internal version param to below extensions
only
APP_VERSION_EXTN = ('.css', '.js', '.html', '.svg', '.png', '.gif',
'.ico')

#####
# Server Connection Driver
Settings
#####

# The default driver used for making connection with
PostgreSQL
PG_DEFAULT_DRIVER = 'psycopg2'

# Maximum allowed idle time in minutes before which releasing the
connection
# for the particular session. (in
minutes)
MAX_SESSION_IDLE_TIME =
60

#####
# User account and settings
storage
#####

# The default path to the SQLite database used to store user accounts
and
# settings. This default places the file in the same directory as
this
# config file, but generates an absolute path for use throughout the
app.
SQLITE_PATH = env('SQLITE_PATH') or os.path.join(DATA_DIR,
'pem.db')

# SQLITE_TIMEOUT will define how long to wait before throwing the error
-
# OperationError due to database lock. On slower system, you may need to
change
# this to some higher
value.
# (Default: 500
milliseconds)
SQLITE_TIMEOUT =
500

# Allow database connection passwords to be saved if the user
chooses.
# Set to False to disable password
saving.
ALLOW_SAVE_PASSWORD =
True

# Maximum number of history queries stored per
user/server/database
MAX_QUERY_HIST_STORED =
20

#####
# Server-side session storage path
#
# SESSION_DB_PATH (Default:
$HOME/.pem7/sessions)
#####
#

```

```

# We use SQLite for server-side session storage. There will be
one
# SQLite database object per session
created.
#
# Specify the path used to store your session
objects.
#
# If the specified directory does not exist, the setup script will
create
# it with permission mode 700 to keep the session database
secure.
#
# On certain systems, you can use shared memory (tmpfs) for
maximum
# scalability, for example, on
Ubuntu:
#
# SESSION_DB_PATH =
'/run/shm/pem7_session'
#
#####
SESSION_DB_PATH = os.path.join(DATA_DIR, 'sessions')

SESSION_COOKIE_NAME =
'pem7_session'

#####
# Mail content
settings
#####
UPGRADE_CHECK_ENABLED =
False

# Check if the detected browser is
supported
CHECK_SUPPORTED_BROWSER = True

#####
# Storage Manager storage url config
settings
# If user sets STORAGE_DIR to empty it will show all volumes if
platform
# is Windows, '/' if it is Linux, Mac or any other unix type
system.
#
# For
example:
# 1. STORAGE_DIR = get_drive("C") or get_drive() # return C:/ by
default
# where C can be any drive character such as "D", "E", "G"
etc
# 2. Set path manually
like
# STORAGE_DIR =
"/path/to/directory/"
#####
STORAGE_DIR = os.path.join(DATA_DIR,
'storage')

#####
# Test settings - used primarily by the regression suite, not for
users
#####
# Set default testing
mode
TESTING_MODE = False

```

```

# The default path for SQLite database for
testing
TEST_SQLITE_PATH = os.path.join(DATA_DIR,
'test_pem.db')

#####
# Allows flask application to response to the each request
asynchronously
#####
THREADED_MODE = True

#####
# Default locations for binary utilities (pg_dump, pg_restore
etc)
#
# These are intentionally left empty in the main config file, but
are
# expected to be overridden by packagers in
config_distro.py.
#
# A default location can be specified for each database driver ID,
in
# a dictionary. Either an absolute or relative path can be
specified.
#
# Version-specific defaults can also be specified, which will take
priority
# over un-versioned
paths.
#
# In cases where it may be difficult to know what the working
directory
# is, "$DIR" can be specified. This will be replaced with the path to
the
# top-level pgAdmin4.py file. For example, on macOS we might
use:
#
#
$DIR/../../SharedSupport
#
#####
DEFAULT_BINARY_PATHS =
{
    "pg": "",
    "pg-9.6": "",
    "pg-10": "",
    "pg-11": "",
    "pg-12": "",
    "pg-13": "",
    "ppas": "",
    "ppas-9.6": "",
    "ppas-10": "",
    "ppas-11": "",
    "ppas-12": "",
    "ppas-13": ""
}

#####
# Database config
settings
#####

# Database
Host

```

```

# default:
'127.0.0.1'
#
# PEM_DB_HOST =
'localhost'

# Database
Name
# default:
'pem'
#
# PEM_DB_NAME =
'pem'

# Database
Port
# default:
5432
# PEM_DB_PORT =
5432

# Do not allow SQLALCHEMY to track modification as it is going to
be
# deprecated in
future
#####
SQLALCHEMY_TRACK_MODIFICATIONS = False

#####

#####
# Number of records to fetch in one batch in query tool when query
result
# set is
large.
#####
ON_DEMAND_RECORD_COUNT = 1000

#####
# Allow users to display Gravatar image for their username in Server
mode
#####
SHOW_GRAVATAR_IMAGE =
False

#####
# Set cookie path and
options
#####
COOKIE_DEFAULT_PATH =
 '/'
COOKIE_DEFAULT_DOMAIN =
None
SESSION_COOKIE_DOMAIN =
None
SESSION_COOKIE_SAMESITE = 'Lax'
SESSION_COOKIE_SECURE =
True,
SESSION_COOKIE_HTTPONLY = True,

#####
# Session expiration support
#####
# SESSION_EXPIRATION_TIME is the interval in Days. Session will
be
# expire after the specified number of
*days*.

```

```

SESSION_EXPIRATION_TIME = 1

# CHECK_SESSION_FILES_INTERVAL is interval in Hours. Application will
check
# the session files for cleanup after specified number of
*hours*.
CHECK_SESSION_FILES_INTERVAL = 24

# USER_INACTIVITY_TIMEOUT is interval in Seconds. If the PEM GUI screen
is
# left unattended for <USER_INACTIVITY_TIMEOUT> seconds then the user
will
# be logged out. When set to 0, the timeout will be
disabled.
# If PEM GUI doesn't detect any activity in the time specified (in
seconds),
# the user will be forcibly logged out from PEM GUI
application.
# Set to zero to disable the
timeout.
USER_INACTIVITY_TIMEOUT = 0

# OVERRIDE_USER_INACTIVITY_TIMEOUT when set to True will
override
# USER_INACTIVITY_TIMEOUT when long running queries in the Query
tool
# or Debugger are running. When the queries complete, the inactivity
timer
# will restart in this case. If set to False, user inactivity may
cause
# transactions or in-process debugging sessions to be
aborted.
OVERRIDE_USER_INACTIVITY_TIMEOUT = True

#####
# SSH Tunneling supports only for Python 2.7 and
3.4+
#####
# Enable the ssh tunnel support in the
application.
SUPPORT_SSH_TUNNEL = True
# Allow SSH Tunnel passwords to be saved if the user
chooses.
# Set to False to disable password
saving.
ALLOW_SAVE_TUNNEL_PASSWORD =
False

#####
# PERFORMANCE
DIAGNOSTIC
#####
# PD_CONNECTION_POOL_USER_MIN_CONNECTION is number of minimum
connections
# allowed in the connection pool for performance diagnostic
module.
PD_CONNECTION_POOL_USER_MIN_CONNECTION = 1

# PD_CONNECTION_POOL_USER_MAX_CONNECTION is number of maximum
connections
# allowed in the connection pool for performance diagnostic
module.
PD_CONNECTION_POOL_USER_MAX_CONNECTION = 3

#####

```



```

# Allows PEM GUI application to create session cookies based
on
# IP address, so even if a cookie is stolen, the attacker will not be
able
# to connect to the server using that stolen
cookie.
# Note: This can cause problems when the server is deployed in dynamic
IP
# address hosting environments, such as Kubernetes or behind
load
# balancers. In such cases, this option should be set to
False.
#####
ENHANCED_COOKIE_PROTECTION =
True

#####
# Authentication
Method
#####

# Default setting is
internal
# If anything other than allowed values is set, it will reset
the
# authentication to
'internal'.
#
# Allowed authentication methods supported are: 'internal',
'kerberos'

PEM_AUTH_METHOD = 'internal'

#####
# Kerberos Authentication
Configuration
#####

KRB_APP_HOST_NAME =
DEFAULT_SERVER

# If the default_keytab_name is not set in krb5.conf
or
# the KRB_KTNAME environment variable is not set then, explicitly
set
# the Keytab
file

KRB_KTNAME = '<KRB5_KEYTAB_FILE>'

# After kerberos authentication, user will be added into the SQLite
database
# automatically, if set to
True.
# Set it to False, if user should not be added
automatically,
# in this case Admin has to add the user manually in the SQLite
database.

KERBEROS_CCACHE_DIR = os.path.join(DATA_DIR,
'krbccache')

# If set to False, the realm name from the authenticated user principal
is
# stripped off before being passed as database user
name.

```

```

PEM_USER_KRB_INCLUDE_REALM =
True

    # This parameter is applicable only when determine, 'PEM_AUTH_METHOD'
is
    #
    'kerberos'.
    #
    # It determines to allow a user to connect any database server
using
    # using authentication other than
Kerberos.
    #
    # e.g. If 'ALLOW_DATABASE_CONNECTION_WITHOUT_KERBEROS' is True, it will
allow
    # users to connect any database server using the username &
password.
ALLOW_DATABASE_CONNECTION_WITHOUT_KERBEROS =
False

#####
# Two-factor Authentication
Configuration
#####

    # Set it to True, to enable the two-factor
authentication
MFA_ENABLED =
False

    # Set it to True, to ask the users to register forcefully for
the
    # two-authentication methods on logged-in.
MFA_FORCE_REGISTRATION = False

    # pgAdmin supports Two-factor authentication by either sending an one-time
code
    # to an email, or using the TOTP based application like Google
Authenticator.
MFA_SUPPORTED_METHODS = ["email",
"authenticator"]

    # NOTE: Please set the 'Mail server Configuration' to use 'email' as two-
factor
    # authentication
method.

    # Subject for the email verification
code
    # Default: <APP_NAME> - Verification
Code
    # e.g. Postgres Enterprise Manager - Verification
Code
MFA_EMAIL_SUBJECT = None

#####
# Mail Server
Configuration
#####
    # PEM Server can send an email using two
mechanisms
    # 1. Using the SMTP configurations saved in the PEM
configuration.
    # e.g. smtp_enabled, smtp_username, smtp_password, smtp_server,
smtp_port,
    # smtp_encrypted, etc.

```

```

# 2. Use Flask-Mail to send an email using following
configurations.

# NOTE: These configurations are only used by Two-Factor Authentication to
send
#       an email to the user for the OTP at the
moment.

# Use SMTP configuration from the PEM configuration to send an
email
# (Default: True)
MAIL_USE_PEM_INTERNAL =
True

# Use TLS for sending mail, when used the internal
settings
MAIL_INTERNAL_USE_TLS =
False

# If set it to 'False', following mail configuration will be used by MFA
to
# send the code on user specified email
address.

MAIL_SERVER = 'localhost'    # (Default:
localhost)
MAIL_PORT = 25               # (Default:
25)
MAIL_USE_TLS = False        # (Default:
False)
MAIL_USE_SSL = False        # (Default:
False)
MAIL_USERNAME = None        # (Default: None)
MAIL_PASSWORD = None        # (Default: None)
# Flask-Security overrides Flask-Mail's MAIL_DEFAULT_SENDER setting,
so
# that should be set as
such:
SECURITY_EMAIL_SENDER = 'no-reply@localhost' # (Default: 'no-
reply@localhost')

# Please refer the Flask-Mail documentation for details about
these
# configurations.
# Reference: https://pythonhosted.org/Flask-
Mail/

#####
#
COMPRESSION
#####
COMPRESS_MIMETYPES =
[
    'text/css', 'text/xml', 'application/javascript', 'text/javascript',
    'application/json', 'text/json'
]
COMPRESS_LEVEL =
9
COMPRESS_MIN_SIZE = 50

#####
# PSQL tool
settings
#####
# This will enable PSQL tool in PEM. So user can execute the
commands

```

```

# using PSQL terminal in
pgAdmin.
ENABLE_PSQL =
False

#####
# Email address validation
#####

# flask-security-too will validate email addresses and check
deliverability
# by default. Disable the deliverability check by default, which was the
old
# behaviour in <= v5.3
CHECK_EMAIL_DELIVERABILITY =
False
SECURITY_EMAIL_VALIDATOR_ARGS =
\
    {"check_deliverability":
CHECK_EMAIL_DELIVERABILITY}

#####
# ENABLE_BINARY_PATH_BROWSING setting is used to enable the browse
button
# while selecting binary path for the database server in server
mode.
# In Desktop mode it is always enabled and setting is of no
use.
#####
ENABLE_BINARY_PATH_BROWSING = False

#####
# ENABLE_LOGIN_ERROR_MASKING setting is used to hide to
specific
# login error which can limit the attacker's scope while check
for
# valid username &
password.
# When set to True PEM will display generic error
like
# 'Invalid username or password' instead of error
like
# 'FATAL: password authentication failed for user
"XYZ"'
#####
ENABLE_LOGIN_ERROR_MASKING =
False

#####
# ENABLE_DATA_ACCESS_TOOLS option allows PEM users to access
certain
# tools so that PEM users can access their data from PEM
interface.
# When set to False, PEM will disable following
tools,
# 1. Query tool & View Data
option
# 2. Import/Export utility
# 3. Backup
Utility
# 4. Generate SQL option in Schema diff & ERD diagram
tools
#####
ENABLE_DATA_ACCESS_TOOLS =
True

```

```

#####
# ENABLE_DEBUGGER option allows PEM users to access the Debugger
tool.
# When set to False, PEM will disable Debugger
tool
#####
ENABLE_DEBUGGER = True
#####
# Local config
settings
#####

# Load distribution-specific config
overrides
if os.path.exists(os.path.join(root, 'config_distro.py')):
    try:
        from config_distro import *
    except ImportError:
        pass

if os.path.exists(os.path.join(root, 'config_setup.py')):
    # Configuration created by the configure-pem-
server.sh
    try:
        from config_setup import *
    except ImportError:
        pass

if os.path.exists(os.path.join(root, 'config_local.py')):
    #
    # Load local config
overrides
    try:
        from config_local import *
    except ImportError:
        pass

#####
# Skip storing session in files and cache for specific
paths
#####
SESSION_SKIP_PATHS =
[
    '/misc/ping'
]

```

### 34.1.4 Roles for managing PEM

You can use the **Login/Group Role** dialog to allow a role with limited privileges to access PEM features such as the Audit Manager, Capacity Manager, or SQL Profiler. PEM pre-defined roles allow access to PEM functionality; roles that are assigned membership in these roles can access the associated feature.

The screenshot shows a window titled "Create - Login/Group Role" with a close button (X) in the top right corner. Below the title bar are several tabs: "General", "Definition", "Privileges", "Membership" (which is selected and highlighted with an orange underline), "Parameters", "Security", and "SQL".

Under the "Membership" tab, there is a section labeled "Roles". To the right of this label is a container with two role entries, each consisting of a small icon, the role name, and a checkbox:

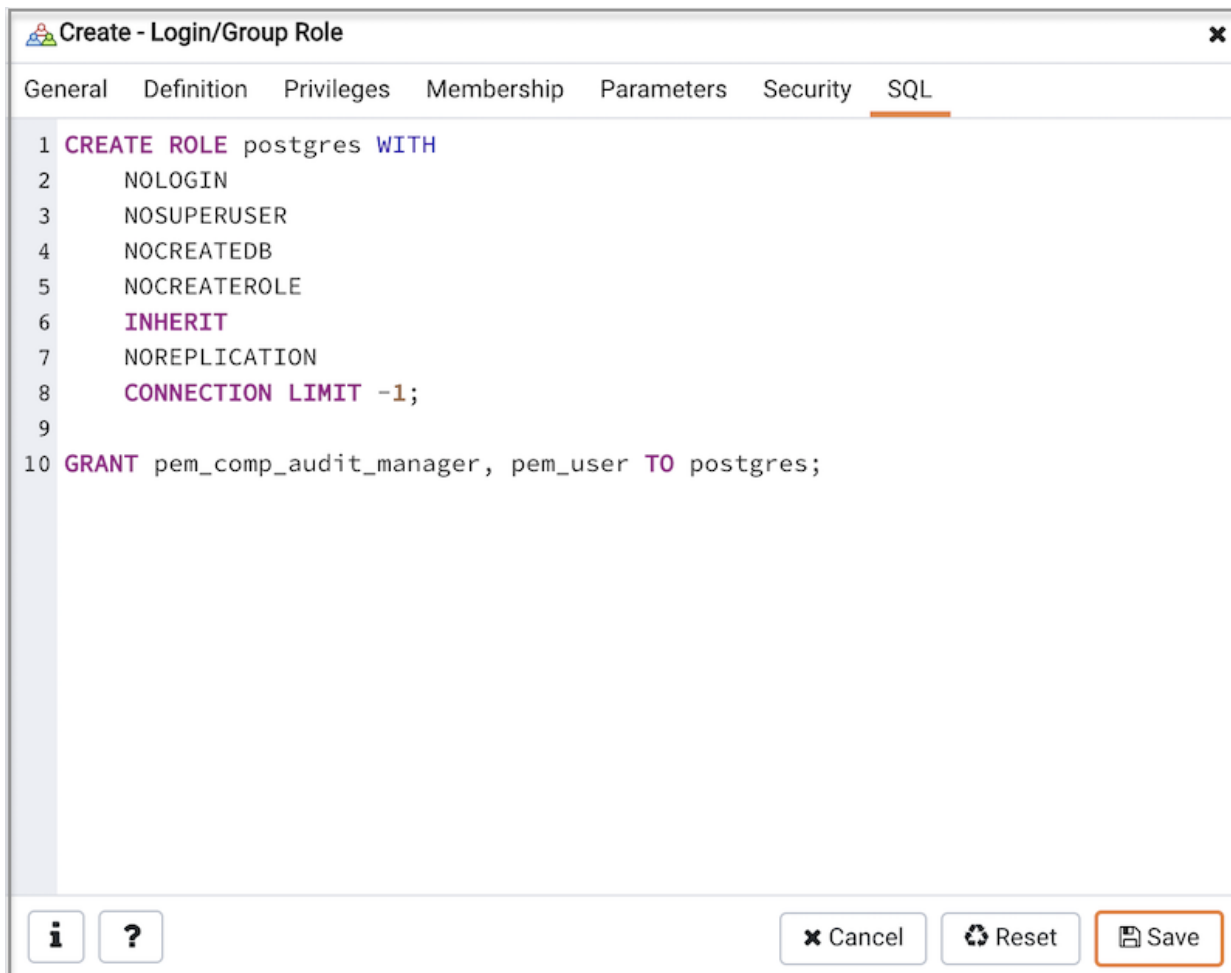
- ☒ pem\_comp\_audit\_manager
- ☒ pem\_user

Below the role entries, there is a text instruction: "Select the checkbox for roles to include WITH ADMIN OPTION." At the bottom of the dialog, there are three buttons: "Cancel" (with an X icon), "Reset" (with a circular arrow icon), and "Save" (with a floppy disk icon and an orange border).

When defining a user, use the **Membership** tab to specify the roles in which the new user is a member. The new user will share the privileges associated with each role in which it is a member. For a user to have access to PEM extended functionality, the role must be a member of the pem\_user role and the pre-defined role that grants access to the feature. Use the **Roles** field to select pre-defined role names from a drop down list.

Check the checkbox to the right of the role name to allow administrative access to the functionality.

The **SQL** tab displays the SQL command that the server will execute when you click **Save**.



The image shows a 'Create - Login/Group Role' dialog box with tabs for General, Definition, Privileges, Membership, Parameters, Security, and SQL. The SQL tab is active, displaying the following SQL code:

```

1 CREATE ROLE postgres WITH
2     NOLOGIN
3     NOSUPERUSER
4     NOCREATEDB
5     NOCREATEROLE
6     INHERIT
7     NOREPLICATION
8     CONNECTION LIMIT -1;
9
10 GRANT pem_comp_audit_manager, pem_user TO postgres;

```

At the bottom of the dialog are buttons for information (i), help (?), Cancel, Reset, and Save.

The examples shown above creates a login role named `postgres` that will have access to the Audit Manager; the role can make unlimited connections to the server at any given time.

You can use PEM pre-defined roles to allow access to the functionality listed in the table below:

| Value                                   | Parent Role  | Description   |
|---|--|---|
| <code>pem_super_admin</code>            |  | Role for administration/management/configuration of all the objects within Postgres Enterprise Manager console.   |
| <code>pem_admin</code>                  | <code>pem_super_admin</code>                               | Role for administration/management/configuration of all the agents, servers, or monitored objects that are visible to a user having <code>pem_admin</code> role. A user with <code>pem_admin</code> role can view and manage only those objects where this role has been mentioned in the Team field under the server's properties. |
| <code>pem_user</code>                   |  | Role for having read-only access to all the agents, servers, or monitored objects that are visible to a user having <code>pem_user</code> role. A user with <code>pem_user</code> role can view only those objects where this role has been mentioned in the Team field under the server's properties.                              |
| <code>pem_config</code>                 | <code>pem_admin</code>                                     | Role for configuration management of Postgres Enterprise Manager.   |
| <code>pem_component</code>              | <code>pem_admin</code>                                     | Role to run/execute all wizard/dialog based components.   |
| <code>pem_rest_api</code>               | <code>pem_admin</code>                                     | Role to access the REST API.  |
| <code>pem_server_service_manager</code> | <code>pem_admin</code>                                     | Role for allowing to restart/reload the monitored database server (if server-id provided).  |
| <code>pem_manage_schedule_task</code>   | <code>pem_admin</code>                                     | Role to configure the schedule tasks.   |
| <code>pem_manage_chart</code>           | <code>pem_admin</code>                                     | Role for managing/configuring custom charts.  |
| <code>pem_manage_alert</code>           | <code>pem_admin</code>                                     | Role for managing/configuring alerts, and its templates.  |
| <code>pem_config_alert</code>           | <code>pem_config</code> ,<br><code>pem_manage_alert</code> | Role for configuring the alerts on any monitored objects.   |

| Value                            | Parent Role                     | Description   |
|----------------------------------|---------------------------------|---|
| pem_manage_probe                 | pem_admin                       | Role to create, update, delete the custom probes, and change custom probe configuration.  |
| pem_config_probe                 | pem_config,<br>pem_manage_probe | Role for probe configuration (history retention, execution frequency, enable/disable the probe) on all visible monitored objects. |
| pem_database_server_registration | pem_admin                       | Role to register a database server.   |
| pem_comp_postgres_expert         | pem_component                   | Role to run the Postgres Expert.  |
| pem_comp_auto_discovery          | pem_component                   | Role to run the Auto discovery of a database server dialog.   |
| pem_comp_log_analysis_expert     | pem_component                   | Role to run the Log Analysis Expert.  |
| pem_comp_sqlprofiler             | pem_component                   | Role to run the SQL Profiler.   |
| pem_manage_efm                   | pem_admin                       | Role to manage Failover Manager functionalities.  |
| pem_comp_capacity_manager        | pem_component                   | Role to run the Capacity Manager.   |
| pem_comp_log_manager             | pem_component                   | Role to run the Log Manager.  |
| pem_comp_audit_manager           | pem_component                   | Role to run the Audit Manager.  |
| pem_comp_tuning_wizard           | pem_component                   | Role to run the Tuning Wizard.  |
| pem_comp_bart                    | pem_component                   | Role to configure and manage BART server.   |

**Note**

The difference between pem\_admin role and pem\_super\_admin role is that a user with pem\_admin role can view and manage only those objects where the role has been mentioned in the Team field under the server's properties, while a user with pem\_super\_admin role can view and manage all the objects within Postgres Enterprise Manager console.

### 34.1.5 The Group Dialog

Use the **Group** dialog to add a new group to the PEM client tree control. You can use a group to simplify management of related servers or agents.

Use the **Name** field to specify a name that will identify the group in the **PEM** browser tree control.

- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

To add a server or agent to a group, right-click on the name of a server or agent, and select **Properties...** to open the properties dialog. Then, use the drop-down listbox in the **Group** field to select the group in which the object should reside.

### 34.1.6 Automatic Discovery of Servers



Use the **Auto Discovery** dialog to instruct a PEM agent to locate database servers that reside on a monitored system, and add a binding that allows the agent to monitor the selected server.

To enable auto discovery for a specific agent, you must enable the **Server Auto Discovery** probe. To access the **Manage Probes** tab, highlight the name of a PEM agent in the PEM client tree control, and select **Manage Probes...** from the **Management** menu. When the Manage Probes tab opens, confirm that the slider control in the **Enabled?** column is set to **Yes**.

To open the **Auto Discovery** dialog, highlight the name of a PEM agent in the PEM client tree control, and select **Auto Discovery...** from the **Management** menu.

When the **Auto Discovery** dialog opens, the **Discovered Database Servers** box will display a list of servers that are not currently monitored by a PEM agent. Check the box next to a server name to display information about the server in the **Server Connection Details** box, and provide any missing information to bind the server to the currently selected agent in the **Agent Connection Details** box.

Use the **Select All** button to select the box next to all of the displayed servers, or **Unselect All** to unselect all of the boxes to the left of the server names.

The fields in the **Server Connection Details** box provide information about the server that PEM will monitor:

- Accept or modify the name of the monitored server in the **Name** field. The specified name will be displayed in the tree control of the PEM client.
- Use the **Server group** drop-down listbox to select the server group under which the server will be displayed in the PEM client tree control.
- Use the **Host name/address** field to specify the IP address of the monitored server.
- The **Port** field displays the port that is monitored by the server; this field may not be modified.
- Provide the name of the service in the **Service ID** field. Please note that the service name must be provided to enable some PEM functionality.
- By default, the **Maintenance database** field indicates that the selected server uses a **postgres** maintenance database. Customize the content of the **Maintenance database** field for your installation.

The fields in the **Agent Connection Details** box specify the properties that the PEM agent will use when connecting to the server:

- The **Host** field displays the IP address that will be used for the PEM agent binding.
- The **Username** field displays the name that will be used by the PEM agent when connecting to the selected server.
- The **Password** field displays the password associated with the specified user name.
- Use the drop-down listbox in the **SSL mode** field to specify your SSL connection preferences.

When you've finished specifying the connection properties for the servers that you are binding for monitoring, click the **OK** button to save the properties. Click **Cancel** to exit without preserving any changes.

**Auto Discovery**

The servers below were discovered by the agent on this system. Check the ones you wish to bind to the agent, and select each in turn to enter the configuration details.

**Discovered Database Servers**

- ☒ PostgreSQL 9.6 database server
- ☐ PostgreSQL 12 database server

☒ Select All ☐ Unselect All

**Server Connection Details**

Enter the connection details that the PEM Server will use to connect to the database server.

|                      |                                |
|----------------------|--------------------------------|
| Name                 | PostgreSQL 9.6 database server |
| Server group         | PEM Server Directory           |
| Host name/address    | 127.0.0.1                      |
| Port                 | 5433                           |
| Service ID           | postgresq-9.6.service          |
| Maintenance database | postgres                       |

Enter the connection details that the PEM Agent will use to connect to the database server.

**Agent Connection Details**

|          |           |
|----------|-----------|
| Host     | 127.0.0.1 |
| Username | postgres  |
| Password | ...       |
| SSL mode | Allow     |

The dialog shown above displays the values required to bind an instance of Advanced Server for monitoring by PEM.

### 34.1.7 Defining a server

Use the **Create - Server** dialog to describe a new server connection, bind the server to a PEM agent, and display the server to the PEM tree control.

**Create - Server**

General Connection SSL SSH Tunnel Advanced PEM Agent BART

**Name** EPAS 12

**Group** PEM Server Directory

**Team**

**Background** ☐

**Foreground** ☐

**Connect now?** ☒

**Comments**

Use the fields on the **General** tab to describe the general properties of the server:

- Use the **Name** field to specify a user-friendly name for the server. The name specified will identify the server in the PEM client tree control.
- You can use **groups** to organize your servers and agents in the PEM client tree control. Using groups can help you manage large numbers of servers more easily. For example, you may want to have a production group, a test group, or LAN specific groups. Use the **Group** drop-down listbox to

select the group in which the new server will be displayed.

- Use the **Team** field to specify a PostgreSQL role name. Only PEM users who are members of this role, who created the server initially, or have superuser privileges on the PEM server will see this server when they logon to PEM. If this field is left blank, by default all PEM users will see the server. You can use the `show_objects_with_no_team` parameter in the **Server Configuration** dialog to change the behaviour. If `show_objects_with_no_team` is set to false, the server with no team will not be visible to all other users.
- Use the **Background** color selector to select the color that will be displayed in the PEM tree control behind database objects that are stored on the server.
- Use the **Foreground** color selector to select the font color of labels in the PEM tree control for objects stored on the server.
- Check the box next to **Connect now?** to instruct PEM to attempt a connection to the database server when you click the Save button on the Create - Server dialog. Leave the **Connect now?** checkbox unchecked if you do not want to establish a connection to the server immediately. If you do not select the **Connect now?** option, the connection parameters are not validated until you attempt a connection.
- Provide notes about the server in the **Comments** field.

**Create - Server**

General **Connection** SSL SSH Tunnel Advanced PEM Agent BART

Host name/address: localhost

Port: 5432

Maintenance database: postgres

Kerberos Authentication? ☒ Yes

Username: shubham@SHUBHAM.ENTERPRISEDB

'shubham@SHUBHAM.ENTERPRISEDB' will be used as the database username, when 'Kerberos Authentication?' is set to 'Yes'

Password:

Save password? ☐

Role:

Use fields on the **Connection** tab to specify connection details for the server:

- Specify the IP address of the server host, or the fully qualified domain name in the **Host name/address** field. On Unix based systems, the address field may be left blank to use the default PostgreSQL Unix Domain Socket on the local machine, or may be set to an alternate path containing a PostgreSQL socket. If you enter a path, the path must begin with a "/".
- Specify the port number of the host in the **Port** field.
- Use the **Maintenance database** field to specify the name of the initial database that PEM will connect to, and that will contain the **pgAgent** schema and admin pack objects if installed (both are optional). If you are planning to monitor PGD via PEM, then specify the PGD-enabled database name in this field. On an Advanced Server database, the maintenance database is named 'edb'. PostgreSQL 8.1 and above, the maintenance DB for PostgreSQL is named 'postgres'; on earlier versions, `template1` is often used, though it is preferable to create a `postgres` database for this purpose to avoid cluttering the template database.
- Set the **Kerberos Authentication** to **Yes** to use the Kerberos Authentication for a monitored server. By default, the monitored server uses the same authentication method as the PEM Server. If the monitored server does not want to use Kerberos Authentication then `ALLOW_DATABASE_CONNECTION_WITHOUT_KERBEROS` must be set to `TRUE` in `config_local.py` file.
- Specify the name that will be used when authenticating with the server in the **Username** field. If the **Kerberos Authentication** method is used then the username field will be populated automatically and will be disabled.
- Provide the password associated with the specified user in the **Password** field. If the **Kerberos Authentication** method is used then this field is disabled.
- Check the box next to **Save password?** to instruct the PEM server to save the password in encrypted format on the PEM server backend database server for later reuse. Password will be stored per server per user basis, hence - it won't be shared with other team members. To remove a password, disconnect from the server, click on the 'Clear Saved Password' menu item under Object/Context menu of the database server.
- Use the **Role** field to specify the name of the role that is assigned the privileges that the client should use after connecting to the server. This allows you to connect as one role, and then assume the permissions of another role when the connection is established (the one you specified in this field). The connecting role must be a member of the role specified.

**Create - Server**

General Connection **SSL** SSH Tunnel Advanced PEM Agent BART

SSL mode: Prefer

Client certificate: ...

Client certificate key: ...

Root certificate: ...

Certificate revocation list: ...

SSL compression?: No

Cancel Reset Save

Use the fields on the **SSL** tab to configure SSL.

- Use the drop-down list box in the **SSL mode** field to select the type of SSL connection the server should use. For more information about using SSL encryption, see [Section 34.19 of the Postgres documentation](#).

You can use the platform-specific File manager dialog to upload files that support SSL encryption to the server. To access the File manager dialog, click the icon that is located to the right of each of the following fields.

- Use the **Client certificate** field to specify the file containing the client SSL certificate. This file will replace the default `<STORAGE_DIR>/<USERNAME>/.postgresql/postgresql.crt`. This parameter is ignored if an SSL connection is not made.
- Use the **Client certificate key** field to specify the file containing the secret key used for the client certificate. This file will replace the default `<STORAGE_DIR>/<USERNAME>/.postgresql/postgresql.key` if PEM is installed in Web mode. This parameter is ignored if an SSL connection is not made.
- Use the **Root certificate** field to specify the file containing the SSL certificate authority. This file will replace the default `~/.postgresql/root.crt`. This parameter is ignored if an SSL connection is not made.
- Use the **Certificate revocation list** field to specify the file containing the SSL certificate revocation list. This list will replace the default list, found in `~/.postgresql/root.crl`. This parameter is ignored if an SSL connection is not made.
- When **SSL compression?** is set to **True**, data sent over SSL connections will be compressed. The default value is **False** (compression is disabled). This parameter is ignored if an SSL connection is not made.

**WARNING:** The certificates, private keys, and the revocation list are stored in the per-user file storage area on the server, which is owned by the user account under which the PEM server process is run. This means that administrators of the server may be able to access those files; appropriate caution should be taken before choosing to use this feature.

**Create - Server**

General Connection SSL **SSH Tunnel** Advanced PEM Agent BART

Use SSH tunneling ☐ No

Tunnel host

Tunnel port

Username

Authentication ☐ Password

Identity file  \*\*\*

Password

Save password? ☐

Cancel Reset Save

Use the fields on the **SSH Tunnel** tab to configure SSH Tunneling. You can use a tunnel to connect a database server (through an intermediary proxy host) to a server that resides on a network to which the client may not be able to connect directly.

- Set **Use SSH tunneling** to **Yes** to specify that PEM should use an SSH tunnel when connecting to the specified server.
- Specify the name or IP address of the SSH host (through which client connections will be forwarded) in the **Tunnel host** field.
- Specify the port of the SSH host (through which client connections will be forwarded) in the **Tunnel port** field.
- Specify the name of a user with login privileges for the SSH host in the **Username** field.
- Specify the type of authentication that will be used when connecting to the SSH host in the **Authentication** field.
  - Select **Password** to specify that PEM will use a password for authentication to the SSH host. This is the default.
  - Select **Identity file** to specify that PEM will use a private key file when connecting.
- If the SSH host is expecting a private key file for authentication, use the **Identity file** field to specify the location of the key file.
- If the SSH host is expecting a password, use the **Password** field to specify the password, or if an identity file is being used, the passphrase.

**Create - Server**

General Connection SSL SSH Tunnel **Advanced** PEM Agent BART

Host address

DB restriction

Password file

Service ID

EFM cluster name

EFM installation path

Connection timeout (seconds) 10

Cancel Reset Save

Use fields on the **Advanced** tab to specify details that are used to manage the server:

- Specify the IP address of the server host in the **Host Address** field.
- Use the **DB restriction** field to specify a SQL restriction that will be used against the `pg_database` table to limit the databases displayed in the tree control. For example, you might enter: `'live_db', 'test_db'` to instruct the PEM browser to display only the `live_db` and `test_db` databases.
- Use the **Password file** field to specify the location of a password file (`.pgpass`). The `.pgpass` file allows a user to login without providing a password when they connect, and it must be present on the PEM server. For more information, see [Section 34.16 of the Postgres documentation](#). Please note: Use of a password file is only supported when PEM is using libpq v10.0 or later to connect to the server.
- Use the **Service ID** field to specify parameters to control the database service process. For servers that are stored in the Enterprise Manager directory, enter the service ID. On Windows machines, this is the identifier for the Windows service. On \*nix machines, this is the name of the init script used to start the server in `/etc/init.d`. An example of an ID on all platforms is `postgresql-9.0`. For local servers, the setting is operating system dependent:
  - If the PEM client is running on a Windows machine, it can control the postmaster service if you have enough access rights. Enter the name of the service. In case of a remote server, it must be prepended by the machine name (e.g. `PSE1\pgsql-8.0`). PEM will automatically discover services running on your local machine.
  - If the PEM client is running on a Unix machine, it can control processes running on the local machine if you have enough access rights. Enter a full path and needed options to access the `pg_ctl` program. When executing service control functions, PEM will append `status/start/stop` keywords to this. For example: `sudo /usr/local/pgsql/bin/pg_ctl -D /data/pgsql`
- If the server is a member of a **Failover Manager** cluster, you can use PEM to monitor the health of the cluster and to replace the primary node if necessary. To enable PEM to monitor Failover Manager, use the **EFM cluster name** field to specify the cluster name. The cluster name is the prefix of the name of the Failover Manager cluster properties file. For example, if the cluster properties file is named `efm.properties`, the cluster name is `efm`.
- If you are using PEM to monitor the status of a **Failover Manager** cluster, use the **EFM installation path** field to specify the location of the Failover Manager binary file. By default, the Failover Manager binary file is installed in `/usr/efm-2.x/bin`, where `x` specifies the Failover Manager version.

Postgres Enterprise Manager Server

General

Connection

SSL

SSH Tunnel

Advanced

PEM Agent

BART

Connection Parameters

Advanced

Bound agent

Postgres Enterprise Manager Host

x

Remote monitoring?

No

Host

127.0.0.1

Port

5444

SSL

Disable

Database

postgres

Username

enterprisedb

Password

Confirm password

i

?

Cancel

Reset

Save

Use fields on the `PEM Agent` tab to specify connection details for the PEM agent:

On `Connection Parameters` tab

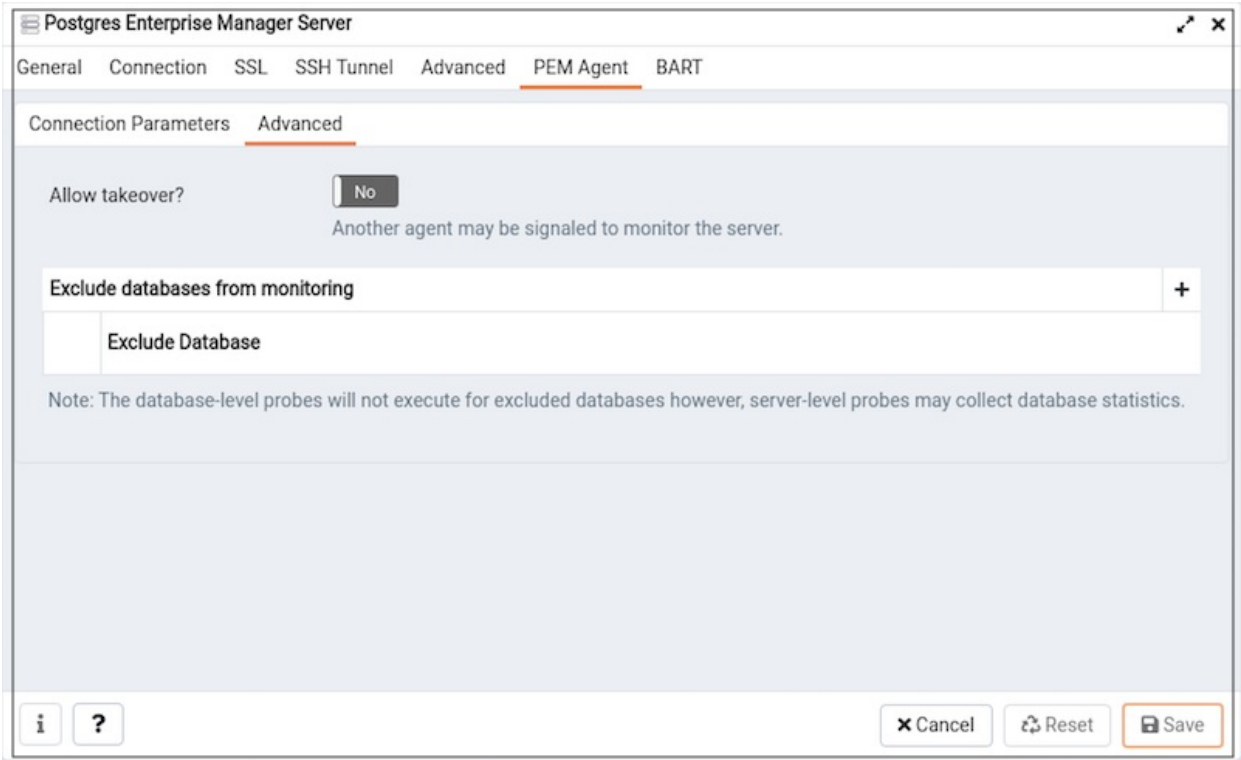
- Specify `Yes` in the `Remote monitoring?` field to indicate that the PEM agent does not reside on the same host as the monitored server. When remote monitoring is enabled, agent level statistics for the monitored server will not be available for custom charts and dashboards, and the remote server will not be accessible by some PEM utilities (such as Audit Manager, Capacity Manager, Log Manager, Postgres Expert and Tuning Wizard).
- Use the drop-down listbox to the right of the `Bound agent` label to select a PEM agent . One agent can monitor multiple Postgres servers.
- Enter the IP address or socket path that the agent should use when connecting to the database server in the `Host` field. By default, the agent will use the host address shown on the `General` tab. On a Unix server, you may wish to specify a socket path, e.g. `/tmp` .
- Enter the `Port` number that the agent will use when connecting to the server. By default, the agent will use the port defined on the `Properties` tab.
- Use the drop-down listbox in the `SSL` field to specify an SSL operational mode; select from require, prefer, allow, disable, verify-ca or verify-full.

| Mode        | Description:   |
|-------------|--|
| require     | To require SSL encryption for transactions between the server and the agent.   |
| prefer      | To use SSL encryption between the server and the agent if SSL encryption is available.   |
| allow       | To allow the connection to use SSL if required by the server.  |
| disable     | To disable SSL encryption between the agent and the server.  |
| verify-ca   | To require SSL encryption, and to require the server to authenticate using a certificate registered by a certificate authority.                      |
| verify-full | To require SSL encryption, and to require the server to authenticate using a certificate registered by a <code>trusted</code> certificate authority. |

For information about using SSL encryption, see [Section 34.19 of the Postgres documentation](#).

- Use the `Database` field to specify the name of the Postgres database to which the agent will initially connect. If you are registering a PGD Node then specify the PGD-enabled database name in this field.
- Specify the name of the user that agent should use when connecting to the server in the `User name` field. Note that if the specified user is not a database superuser, then some of the features will not work as expected. If you are using Postgres version 10 or above, you can use the `pg_monitor` role to grant the required privileges to a non-superuser. For information about `pg_monitor` role, see [Default Roles](#). If you are using Kerberos Authentication method then specify the user having `pgd_monitor` or `pgd_superuser` role in this field.
- Specify the password that the agent should use when connecting to the server in the `Password` field, and verify it by typing it again in the

`Confirm password` field. If you do not specify a password, you will need to configure the authentication for the agent manually; you can use a `.pgpass` file for example.



On `Advanced` tab

- Specify `Yes` in the `Allow takeover?` field to specify that another agent may be signaled (for example, by a fencing script) to monitor the server. This feature allows an agent to take responsibility for the monitoring of the database server if, for example, the server is part of a [high availability](#) failover process.
- Use the `+` sign to add the database you want to exclude from the PEM Monitoring. You cannot exclude the database mentioned on the `Connection Parameters` tab of the `PEM Agent` tab.

**Note**

The database-level probes do not execute for excluded databases, but the server-level probes may collect the database statistics.

If you experience connection problems, please visit the [connection problems](#) page.

To view the properties of a server, right-click on the server name in the PEM client tree control, and select the `Properties...` option from the context menu. To modify a server's properties, disconnect from the server before opening the `Properties` dialog.

### 34.1.8 Defining and Monitoring Postgres instances on AWS

There are two scenarios in which you can monitor a Postgres instance on an AWS host with PEM:

- Postgres Instance running on AWS EC2
- Postgres Instance running on AWS RDS

#### Monitoring a Postgres Instance Running on AWS EC2



After creating a Postgres instance on AWS EC2, you can use the PEM server to register and monitor your instance. The following scenarios are currently supported:

- Postgres instance and PEM Agent running on the same AWS EC2 and a PEM Server running on your local machine.
- Postgres instance and PEM Agent running on the same local machine and a PEM Server running on AWS EC2.
- Postgres instance and PEM Agent running on the same AWS EC2 and a PEM Server running in different AWS EC2.

#### Note

In the first two scenarios, you must configure the VPN on AWS EC2 , so the AWS EC2 instance can access the `pem` database. Please contact your network administrator to setup the VPN if needed.

The PEM Agent running on AWS EC2 or on your local machine should be registered to the PEM Server. Please note that when registering the PEM Agent with the PEM Server you should use the hostname of AWS EC2 instance. For more details on registering the PEM Agent see, [PEM Self Registration](#).

You can register the Postgres instance running on AWS EC2 on PEM Server using the `Create - Server` dialog. For more details on registering the server using `Create - Server` dialog see, [Define a Server](#). Use the `PEM Agent` tab on the `Create - Server` dialog to bind the registered PEM Agent with the Postgres instance.

When the PEM Agent is registered to the PEM Server and your Postgres instance that is running on AWS EC2 is registered to the PEM Server, you can monitor your instance with PEM.

## Monitoring a Postgres Instance Running on AWS RDS

While creating an AWS RDS database, choose `PostgreSQL` when prompted for `Engine options` . After creating a `Postgres(RDS)` instance on AWS, use `Create - Server` dialog to add the `Postgres(RDS)` instance to the PEM Server. Using this dialog you can describe a new server connection, bind the server to a PEM Agent, and display the server to the PEM browser tree control.

For detailed information on the `Create - Server` dialog and configuration details for each tab, see [Define a Server](#).

The `PEM Agent` tab in the `Create - Server` dialog must have the `Remote Monitoring` field set to `Yes` to monitor the `Postgres(RDS)` instance on AWS instance using PEM Server.

The screenshot shows a dialog box titled "aws\_rds\_pgsql\_12" with a close button (X) in the top right corner. The dialog has several tabs: General, Connection, SSL, SSH Tunnel, Advanced, PEM Agent (selected), and BART. The "PEM Agent" tab contains the following fields and controls:

- Bound agent:** A dropdown menu showing "Postgres Enterprise Manager Host" with a clear button (X) and a dropdown arrow.
- Remote monitoring?:** A toggle switch set to "Yes".
- Host:** A text input field containing "54.157.22.204".
- Port:** A text input field containing "5432".
- SSL:** A dropdown menu set to "Disable".
- Database:** A text input field containing "postgres".
- Username:** A text input field containing "postgres".
- Password:** A text input field.
- Confirm password:** A text input field.
- Allow takeover?:** A toggle switch set to "No".

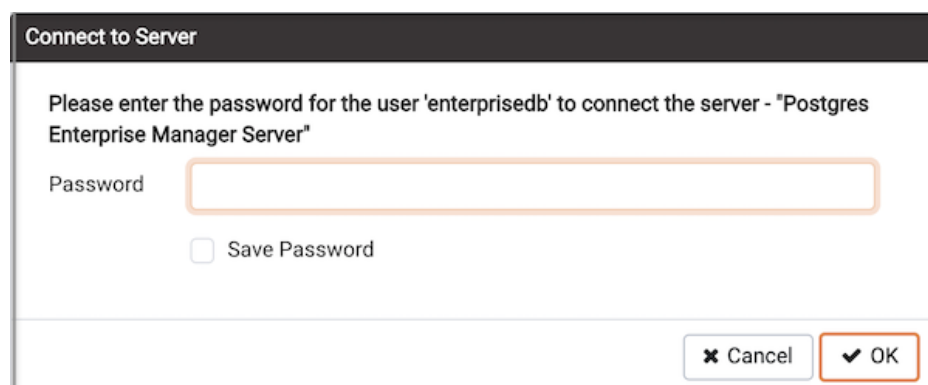
At the bottom of the dialog, there are three buttons: "Cancel", "Reset", and "Save".

As the PEM Agent will be monitoring the Postgres(RDS) AWS instance remotely, the functionality will be limited as described below:

| Feature Name                 | Works with remote PEM Agent | Comments   |
|------------------------------|-----------------------------|--|
| Audit Manager                | No                          |  |
| Capacity Manager             | Limited                     | There will be no correlation between the database server and operating system metrics.   |
| Log Manager                  | No                          |  |
| Manage Alerts                | Limited                     | When you run an alert script on the database server, it will run on the machine where the bound PEM Agent is running, and not on the actual database server machine.               |
| Manage Charts                | Yes                         |  |
| Manage Dashboards            | Limited                     | Some dashboards may not be able to show complete data. For example, the operating system information of the database server will not be displayed as it is not available.          |
| Manage Probes                | Limited                     | Some of the PEM probes will not return information, and some of the functionalities may be affected. For details about probe functionality, see <a href="#">Agent privileges</a> . |
| Postgres Expert              | Limited                     | The Postgres Expert will provide partial information as operating system information is not available.   |
| Postgres Log Analysis Expert | No                          | The Postgres Log Analysis Expert will not be able to perform an analysis as it is dependent on the logs imported by log manager, which will not work as required.                  |
| Scheduled Tasks              | Limited                     | Scheduled tasks will work only for database server; scripts will run on a remote Agent.  |
| Tuning Wizard                | No                          |  |
| System Reports               | Yes                         |  |
| Core Usage Reports           | Limited                     | The Core Usage report will not show complete information. For example, the platform, number of cores, and total RAM will not be displayed.   |
| Managing BART                | No                          | BART requires password less authentication between two machines, where database server and BART are installed. An AWS RDS instance doesn't allow to use host access.               |

### 34.1.9 Connect to server

After defining a server connection, use the **Connect to Server** dialog to authenticate with a server and access the objects stored on the server. To access the dialog, right click on the server name in the PEM client tree control, and select **Connect Server** from the context menu.



The image shows a 'Connect to Server' dialog box. It has a title bar 'Connect to Server'. Inside, it says 'Please enter the password for the user 'enterprisedb' to connect the server - "Postgres Enterprise Manager Server"'. There is a text input field for the password. Below the field is a checkbox labeled 'Save Password'. At the bottom right, there are two buttons: 'Cancel' and 'OK'.

If prompted, provide authentication information for the selected server:

- Use the **Password** field to provide the password of the user that is associated with the defined server.
- Check the box next to **Save Password** to instruct the server to save the password for future connections; if you save the password, you will not be prompted when reconnecting to the database server with this server definition.

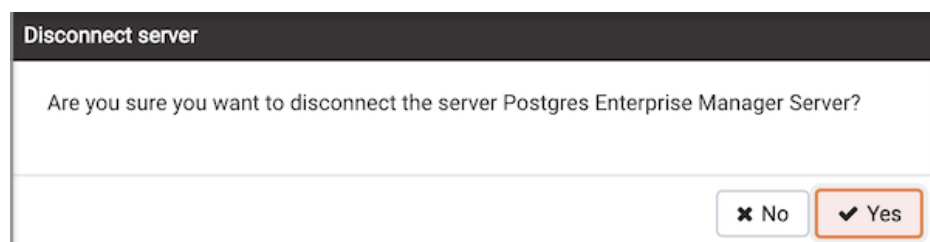
The browser displays a message in a green status bar in the lower right corner when the server connects successfully.

If you receive an error message while attempting a connection, verify that your network is allowing PEM and the host of the database server to communicate. For detailed information about a specific error message, please see the [Connection Error](#) help page.

To review or modify connection details, right-click on the name of the server, and select **Properties...** from the context menu.

## Disconnecting from a Server

To disconnect from a server, right-click on the server name in the **Browser** tree control and select Disconnect Server from the context menu. A popup will ask you to confirm that you wish to disconnect the selected server.



### 34.1.10 Controlling a Server

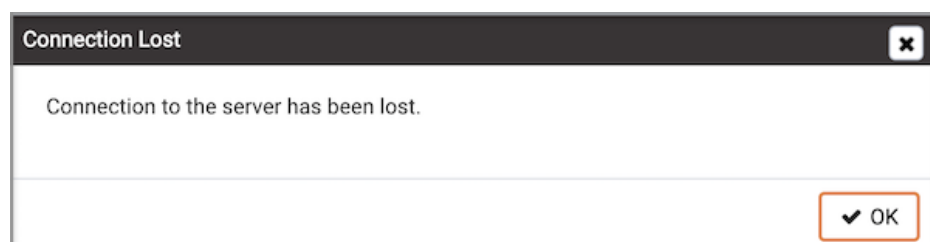
If you provided a **Service ID** on the **Advanced** tab of the **Server** property dialogue, the PEM server can control the database service process.

- If the PEM client is running on a Windows machine, it can control the postmaster service if you have sufficient access rights. In case of a remote server, the service name must be prepended by the machine name (e.g. PSE1pgsql-8.0).
- If the PEM client is running on a Unix machine, it can control processes running on the local machine if you have sufficient access rights. When executing service control functions, PEM will append status/start/stop keywords to the service name provided.

### 34.1.11 Connection error

When connecting to a PostgreSQL server, you may get an error message. If you encounter an error message, please review the message carefully; each error message attempts to incorporate the information you'll need to resolve the problem. For more details about specific errors, please locate the error message in the list below:

#### Connection to the server has been lost



This error message indicates that the connection attempt has taken longer than the specified threshold; there may be a problem with the connection properties provided on the **Server** dialog, network connectivity issues, or the server may not be running.

could not connect to Server: Connection refused

**Connect to Server**

Please enter the password for the user 'enterprisedb' to connect the server - "Postgres Enterprise Manager Server"

Password

☐ Save Password

**⚠** could not connect to server: Connection refused Is the server running on host "127.0.0.1" and accepting TCP/IP connections on port 5445?

If PEM displays this message, there are two possible reasons for this:

- the database server isn't running - simply start it.
- the server isn't configured to accept TCP/IP requests on the address shown.

For security reasons, a PostgreSQL server "out of the box" doesn't listen on TCP/IP ports. Instead, it must be enabled to listen for TCP/IP requests. This can be done by adding `tcpip = true` to the `postgresql.conf` file for Versions 7.3.x and 7.4.x, or `listen_addresses=*` for Version 8.0.x and above; this will make the server accept connections on any IP interface.

For further information, please refer to the PostgreSQL documentation about [runtime configuration](#).

**FATAL: no pg\_hba.conf entry**

**Connect to Server**

Please enter the password for the user 'postgres' to connect the server - "PG 9.5"

Password

☐ Save Password

**⚠** FATAL: no pg\_hba.conf entry for host "127.0.0.1", user "postgres", database "postgres", SSL off

If PEM displays this message when connecting, your server can be contacted correctly over the network, but is not configured to accept your connection. Your client has not been detected as a legal user for the database.

To connect to a server, the `pg_hba.conf` file on the database server must be configured to accept connections from the host of the PEM client. Modify the `pg_hba.conf` file on the database server host, and add an entry in the form:

- `host template1 postgres 192.168.0.0/24 md5` for an IPV4 network
- `host template1 postgres ::ffff:192.168.0.0/120 md5` for an IPV6 network

For more information, please refer to the PostgreSQL documentation about [client authentication](#).

**FATAL: password authentication failed**

- The `password authentication failed for user` error message indicates there may be a problem with the password you entered. Retry the password to confirm you entered it correctly. If the error message returns, make sure that you have the correct password, that you are authorized to access the server, and that the access has been correctly configured in the server's `postgresql.conf` configuration file.

### 34.1.12 Enable Kerberos Authentication

PEM 8.1 onwards supports `Kerberos Authentication`. So you need to set up the Kerberos authentication at the time of configuring the PEM Server. For more details, see [PEM Installation docs](#).

#### Principal and Keytab file for PEM Webserver (apache/HTTP) and PEM Backend Database

- Add the `principal` and generate the `keytab` file for `PEM webserver` (apache/HTTP) and copy it to the machine where the PEM webserver resides. Ensure that the operating system user owning the `PEM Webserver` is the owner of this file and accessible by that user.
- Add the `principal` and generate the `keytab` file for the `PEM backend database` server and copy it to the machine where the PEM backend database resides. Ensure that the operating system user owning the `PEM backend database` server data directory is the owner of this file and accessible by that user.

#### Configuration settings

To enable Kerberos authentication for PEM, you must configure the following parameters in `config_setup.py` on the system where the PEM Server is installed.

- `PEM_APP_HOST` - Specify the name of the `PEM Webserver Hostname`. Please note that if it is not set, then it will take the value of the `default_server` parameter.
- `PEM_AUTH_METHOD` - Set the value to `kerberos` to use kerberos authentication.

#### Browser settings to configure Kerberos Authentication

You need to do the browser settings on the client machine to use the `Spnego/Kerberos`.

- For Mozilla Firefox
  - Open the low level Firefox configuration page by loading the `about:config` page.
  - In the Search text box, enter: `network.negotiate-auth.trusted-uris`

- Double-click the `network.negotiate-auth.trusted-uris` preference and enter the hostname or the domain of the web server that is protected by Kerberos HTTP SPNEGO. Separate multiple domains and hostnames with a comma.
- Click OK.

- For Google Chrome

- For Windows:

- Open the Control Panel to access the Internet Options dialog.
- Select the Security tab.
- Select the Local Intranet zone and click the Sites button.
- Make sure that the first two options, Include all local (intranet) sites not listed in other zones and Include all sites that bypass the proxy server are checked.
- Click Advanced and add the names of the domains that are protected by Kerberos HTTP SPNEGO, one at a time, to the list of websites. For example, myhost.example.com. Click Close.
- Click OK to save your configuration changes.

- For Linux or MacOS:

- Add the `--auth-server-whitelist` parameter to the google-chrome command. For example, to run Chrome from a Linux prompt, run the google-chrome command as follows:

```
google-chrome --auth-server-whitelist =
"hostname/domain"
```

## 34.2 Managing a PEM Agent

The PEM agent is responsible for implementing scheduled tasks on the PEM server on behalf of the server. The agent runs as a service (on Windows) or as a daemon (on Linux). The PEM server installer automatically installs and configures an agent that is responsible for monitoring the PEM server; you can use the PEM agent installer to add additional agents.

- [PEM Agent Properties](#)
- [Binding an Agent to a Server](#)
- [Controlling the PEM Agent](#)
- [High Availability Integration](#)

The PEM agent has a number of configuration parameters that control agent behaviors and enable PEM functionality. For a list of the parameters, see:

- [PEM Agent Privileges](#)
- [PEM Agent Configuration Parameters](#)
- [PEM Agent Self Registration](#)
- [Register/Unregister database server using PEM Agent](#)

### 34.2.1 PEM Agent Properties

The `PEM Agent Properties` dialog provides information about the PEM agent from which the dialog was opened; to open the dialog, right-click on an agent name in the PEM client tree control, and select `Properties` from the context menu.

Postgres Enterprise Manager Host

General Job Notifications Agent Configurations

Description Postgres Enterprise Manager Host

Group PEM Agents

Team

Heartbeat interval 0 Minutes 30 Seconds

Cancel Reset Save

- The **Description** field displays a modifiable description of the PEM agent. This description is displayed in the tree control of the PEM client.
- You can use **groups** to organize your servers and agents in the PEM client tree control. Use the **Group** drop-down listbox to select the group in which the agent will be displayed.
- Use the **Team** field to specify the name of the group role that should be able to access servers monitored by the agent; the servers monitored by this agent will be displayed in the PEM client tree control to connected team members. Please note that this is a convenience feature. The **Team** field does not provide true isolation, and should not be used for security purposes.
- The **Heartbeat interval** fields displays the length of time that will elapse between reports from the PEM agent to the PEM server. Use the selectors next to the **Minutes** or **Seconds** fields to modify the interval.

The screenshot shows the 'Postgres Enterprise Manager Host' dialog box with the 'Job Notifications' tab selected. The dialog has three tabs: 'General', 'Job Notifications', and 'Agent Configurations'. The 'Job Notifications' tab contains four settings:

- Override default configuration?**: A toggle switch set to 'No'. Description: 'Select to override the default configuration for job notifications. If selected, the following settings will determine whether, when, and which email group will receive the job notification for this agent.'
- Email on job completion?**: A toggle switch set to 'No'. Description: 'Select to receive a notification email on completion of a job (regardless of the result) of this agent.'
- Email on a job failure?**: A toggle switch set to 'No'. Description: 'Select to receive a notification email only on failure of a job of this agent.'
- Email group**: A dropdown menu showing '<Default>'. Description: 'Select the email-group that will receive the notification on completion of a job or scheduled task.'

At the bottom of the dialog are three buttons: 'Cancel', 'Reset', and 'Save' (highlighted with an orange border). There are also information and help icons on the left.

Use the fields on the **Job Notifications** tab to configure the email notification settings on agent level:

- Use the **Override default configuration?** switch to specify if you want the agent level job notification settings to override the default job notification settings. If you select Yes for this switch, you can use the rest of the settings on this dialog to define when and to whom the job notifications should be sent. Please note that the rest of the settings on this dialog work only if you enable the **Override default configuration?** switch.
- Use the **Email on job completion?** switch to specify if the job notification should be sent on the successful job completion.
- Use the **Email on a job failure?** switch to specify if the job notification should be sent on the failure of a job.
- Use the **Email group** field to specify the email group to whom the job notification should be sent.



Postgres Enterprise Manager Host
✕

General
Job Notifications
Agent Configurations

| Parameter          | Value                   | Category      |
|--------------------|-------------------------|---------------|
| Agent Id           | 1                       | configuration |
| Running as root?   | true                    | capability    |
| Running as User    | root                    | capability    |
| Platform           | "CentOS Linux 7 (Core)" | capability    |
| Architecture       | x64                     | capability    |
| PEM host           | 127.0.0.1               | configuration |
| PEM port           | 5444                    | configuration |
| Log level          | debug1                  | configuration |
| Agent SSL key path | /root/.pem//agent1.key  | configuration |
| Agent SSL crt path | /root/.pem//agent1.crt  | configuration |
| Long wait          | 30                      | configuration |
| Short wait         | 10                      | configuration |
| Alert threads      | 1                       | configuration |

i
?
✕ Cancel
🔄 Reset
💾 Save

**Agent Configurations** tab lists down all the current configurations and capabilities of a agent.

- The **Parameter** column displays list of parameters.
- The **Value** column displays current value of the corresponding parameter.
- The **Category** column displays category of the corresponding parameter, it can be either "configuration" or "capability".

## 34.2.2 Binding an Agent to a Server

The PEM agent runs as a service (on Windows) or as a daemon (on Linux), and is responsible for implementing scheduled tasks on the PEM server on behalf of the server. The PEM server installer automatically installs and configures an agent that is responsible for monitoring the PEM server. The PEM agent installer will setup and configure the agent to start automatically at boot time, however the agent can also be manually [started](#) if required.

To create a binding for a registered server, right click on the name of the server in the tree control, and select **Properties** from the context menu. Open the **PEM Agent** tab:

Create - Server

General

Connection

SSL

SSH Tunnel

Advanced

PEM Agent

BART

Bound agent

Postgres Enterprise Manager Host

Remote monitoring?

No

Host

127.0.0.1

Port

5432

SSL

Prefer

Database

postgres

Username

enterprisedb

Password

...

Confirm password

...

Allow takeover?

No

?

?

Cancel

Reset

Save

Use the fields on the **PEM Agent** tab to associate the server (defined on the Connection tab) with a PEM agent:

Use fields on the **PEM Agent** tab to specify connection details for the PEM agent:

- Specify **Yes** in the **Remote monitoring?** field to indicate that the PEM agent does not reside on the same host as the monitored server. When remote monitoring is enabled, agent level statistics for the monitored server will not be available for custom charts and dashboards, and the remote server will not be accessible by some PEM utilities (such as Audit Manager, Capacity Manager, Log Manager, Postgres Expert and Tuning Wizard).
- Select an Enterprise Manager agent using the drop-down listbox to the right of the **Bound agent** label. One agent can monitor multiple Postgres servers.
- Enter the IP address or socket path that the agent should use when connecting to the database server in the **Host** field. By default, the agent will use the host address shown on the **General** tab. On a Unix server, you may wish to specify a socket path, e.g. `/tmp`.
- Enter the **Port** number that the agent will use when connecting to the server. By default, the agent will use the port defined on the **Properties** tab.
- Use the drop-down listbox in the **SSL** field to specify an SSL operational mode; specify require, prefer, allow, disable, verify-ca or verify-full.
- Use the **SSL** field to specify an SSL operational mode.

| Mode        | Specify:   |
|-------------|--|
| require     | To require SSL encryption for transactions between the server and the agent.   |
| prefer      | To use SSL encryption between the server and the agent if SSL encryption is available.   |
| allow       | To allow the connection to use SSL if required by the server.  |
| disable     | To disable SSL encryption between the agent and the server.  |
| verify-ca   | To require SSL encryption, and to require the server to authenticate using a certificate registered by a certificate authority.                |
| verify-full | To require SSL encryption, and to require the server to authenticate using a certificate registered by a <b>trusted</b> certificate authority. |

For information about using SSL encryption, see [Section 31.17 of the Postgres documentation](#).

- Use the **Database** field to specify the name of the Postgres Plus database to which the agent will initially connect.
- Specify the name of the user that agent should use when connecting to the server in the **User name** field. Note that if the specified user is not a database superuser, then some of the features will not work as expected. If you are using Postgres version 10 or above, you can use the **pg\_monitor** role to grant the required privileges to a non-superuser. For information about **pg\_monitor** role, see [Default Roles](#).
- Specify the password that the agent should use when connecting to the server in the **Password** field, and verify it by typing it again in the

`Confirm password` field. If you do not specify a password, you will need to configure the authentication for the agent manually; you can use a `.pgpass` file for example.

- Specify `Yes` in the `Allow takeover?` field to specify that the server may be "taken over" by another agent. This feature allows an agent to take responsibility for the monitoring of the database server if, for example, the server has been moved to another host as part of a [high availability](#) failover process.

### 34.2.3 Controlling the PEM Agent

On Linux platforms, the name of the service script that controls a PEM agent is `pemagent`. You can use the `pemagent` service script to control the PEM agent. Enter:

```
/etc/init.d/pem_agent action
```

Where `action` specifies the action taken by the service. Specify:

- start to start the service.
- stop to stop the service.
- restart to stop and then start the service.
- status to check the status of the service.

To determine if a service is running on RHEL, open a command line, and issue the command:

```
systemctl pemagent action
```

Where `action` is the action taken by the service. You can specify:

- start to start the service.
- stop to stop the service.
- restart to stop and then start the service.
- status to inquire about the current status of the service.

### Controlling the PEM Agent on Windows

The Windows operating system includes a graphical service controller (the Windows `Services` applet) that displays the server status, and offers point-and-click service control. The Services applet can be accessed through the Windows Control Panel. When the utility opens, use the scroll bar to navigate through the listed services to highlight the `Postgres Enterprise Manager - pemAgent` service name.

- Use the Stop the service option to stop a service.
- Use the Pause the service option to instruct Postgres to reload a service's configuration parameters.
- Use the Start the service option to start a service.

### 34.2.4 High Availability Integration

In high availability (HA) configurations, the database servers that are being monitored may be moved ("failed over") to a different host in the event of any problems, such as a hardware failure. There are numerous ways to maintain a backup server using features of Postgres and external tools. Please consult the Postgres documentation for further details.

In order to run in an HA environment, it is recommended that a PEM agent be installed on both the primary host machine, and any secondary machines that may be used as backups. The server is bound to the agent running on the primary host in the [normal fashion](#).

When the clustering solution initiates a failover of Postgres from one server to another, the PEM agent on the server that is taking over the running of the database may be instructed to take over the monitoring of the database server as well. The server must first be configured to allow "takeovers" using the `Allow takeover?` configuration option on the `PEM Agent` tab of the server configuration dialogue.

To instruct the agent to takeover the monitoring of a server, the failover process must simply create a file in a special "flag" directory which will instruct the agent to take responsibility for the specified server. A command such as the following could be added to a failover script on a Linux server for example:

```
touch /tmp/pem/agent-AGENTID/takeover-server-SERVERID
```

where `AGENTID` is the numeric ID of the agent that should takeover the monitoring of the server, and `SERVERID` is the numeric ID of the server that should be taken over. The IDs may be found by logging into the PEM client, and selecting the Agent or Server and viewing the ID values on the `Properties` pane of the main window.

The agent will take over monitoring of the failed-over server within approximately 30 seconds in a standard configuration of PEM.

The flag directory used by the agent is `$TMPDIR/pem/agent-AGENTID` by default (where `$TMPDIR` is as set for the user account under which the agent runs, usually `root` on Linux/Unix, or `Administrator` on Windows). The directory path can be overridden using the `AgentFlagDir` configuration option in the registry on Windows, or the `agent_flag_dir` option in the agent configuration file on other platforms.

### 34.2.5 PEM Agent Privileges

By default, the PEM agent is installed with `root` privileges for the operating system host and superuser privileges for the database server. These privileges allow the PEM agent to invoke unrestricted probes on the monitored host and database server about system usage, retrieving and returning the information to the PEM server.

Please note that PEM functionality diminishes as the privileges of the PEM agent decrease. For complete functionality, the PEM agent should run as `root` and on the same host as the database server.

- If the PEM agent is run under the database server's service account, PEM probes will not have complete access to the statistical information used to generate reports, and functionality will be limited to the capabilities of that account.
- If the PEM agent is run under another lesser-privileged account, functionality will be limited even further.
- If the PEM agent is installed on a different host and is monitoring the database server remotely, then the functionality will be limited.

| Feature Name     | Works with root User | Works with non-root User  | Works with remote PEM Agent  |
|------------------|----------------------|---|--|
| Audit Manager    | yes                  | The Audit Log Manager may be unable to apply requested modifications if the service cannot be restarted. The user running PEM Agent may be different from the user who owns the data directory of the database server, so user running PEM Agent may not be able to change the configuration and also may not be able to restart the services of the database server. | no   |
| Capacity Manager | yes                  | yes   | yes<br><br>NOTE: There will be no co-relation between the database server and operating system metrics |
| Log Manager      | yes                  | The Log Manager may be unable to apply requested modifications if the service cannot be restarted. The user running PEM Agent may be different from the user who owns the data directory of the database server, so user running the PEM Agent may not be able to change the configuration and also may not be able to restart the services of the database server.   | no   |

| Feature Name                 | Works with root User | Works with non-root User   | Works with remote PEM Agent  |
|------------------------------|----------------------|--|--|
|                              |                      |  | yes  |
| Manage Alerts                | yes                  | yes  | NOTE: When run alert script on the database server is selected, it will run on the machine, where bound PEM Agent is running, and not on the actual database server machine. |
| Manage Charts                | yes                  | yes  | yes  |
| Manage Dashboards            | yes                  | Some dashboards may not be able to show complete data. For example, columns such as swap usage, CPU usage, IO read, and IO write will be displayed as 0 in the session activity dashboard.   | Some dashboards may not be able to show complete data. For example, the operating system information of the database server will not be displayed as not available.          |
| Manage Probes                | yes                  | Some of the PEM probes will not return information, and some of functionalities may be affected. For details about probe functionality, see the <a href="#">Agent privileges</a> .   | Some of the PEM probes will not return information, and some of the functionalities may be affected.   |
| Postgres Expert              | yes                  | The Postgres Expert will be able to access the configuration expert and schema expert, but not the security expert.  | The Expert will provide partial information as operating system information is not available.  |
| Postgres Log Analysis Expert | yes                  | The Postgres Log Analysis Expert may not be able to do the analysis as it is dependent on the logs imported by log manager, which will not work as required.   | The Postgres Log Analysis Expert will not be able to do the analysis as it is dependent on the logs imported by log manager, which will not work as required.                |
| Scheduled Tasks              | yes                  | For Linux if user is the same as batch_script_user in agent.cfg then shell script will run.  | Scheduled tasks will work only for database server; scripts will run on a remote Agent.  |
| Tuning Wizard                | yes                  | The Tuning Wizard will be unable to run if the service cannot be restarted. The user running PEM Agent may be different from the user who owns the data directory of the database server, so user running PEM Agent may not be able to change the configuration and also may not be able to restart the services of the database server. | no   |
| System Reports               | yes                  | yes  | yes  |
| Core Usage Reports           | yes                  | yes  | The Core Usage report will not show complete information. For example, the platform, number of cores, and total RAM will not be displayed.                                   |
|                              |                      |  | no   |
| Managing BART                | yes                  | BART and the BART scanner may not be able to start/reload.   | NOTE: BART requires password less authentication between two machines, where database server and BART are installed.   |

## 34.2.6 PEM Agent Configuration Parameters

A number of user-configurable parameters and registry entries control the behavior of the PEM Agent. With the exception of the PEM\_MAXCONN (or pem\_maxconn) parameter, we strongly recommend against modifying any of the configuration parameters or registry entries listed below without first consulting EnterpriseDB support experts.

- On Windows systems, PEM registry entries are located in HKEY\_LOCAL\_MACHINE\Software\EnterpriseDB\PEM\agent
- On Linux systems, PEM configuration options are stored in the agent.cfg file, located (by default) in /usr/edb/pem/agent/etc

| Parameter Name   | Description   | Value (if applicable)  |
|--|---|--|
| PEM_HOST (on Windows) or pem_host (on Linux)                     | The IP address or hostname of the PEM server.   | By default, set to 127.0.0.1.  |
| PEM_PORT (on Windows) or pem_port (on Linux)                     | The database server port to which the agent connects to communicate with the PEM server.  | By default, the PEM server monitors port 5432.   |
| AgentID (on Windows) or agent_id (on Linux)                      | A unique identifier assigned to the PEM agent.  | The first agent is assigned an identifier of '1', the second agent is assigned an identifier of '2', and so on.      |
| AgentUser (on Windows) or agent_user (on Linux)                  | User to connect the PEM database server   | If present, and not set to empty string, it will be used to connect the PEM database server.                         |
| AgentCrtPath (on Windows) or agent_ssl_cert (on Linux)           | The complete path to the PEM agent's certificate file.  | By default, on Windows, C:\Users\user_name\AppData\Roaming\pem\agent.crt. By default on Linux, /root/.pem/agent.crt. |
| AgentKeyPath (on Windows) or agent_ssl_key (on Linux)            | The complete path to the PEM agent's key file.  | By default, on Windows, C:\Users\user_name\AppData\Roaming\pem\agent.key. By default on Linux, /root/.pem/agent.key. |
| AgentFlagDir (on Windows) or agent_flag_dir (on Linux)           | Used for HA support. Specifies the directory path checked for requests to take over monitoring another server. Requests are made in the form of a file in the specified flag directory.   | Not set by default. This option allows you to override the hard-coded default.                                       |
| LogLevel (on Windows) or log_level (on Linux)                    | Log level specifies the type of event that will be written to the PEM log files.  | Log level may be set to: error, debug1, debug2, or warning By default, log level is set to <b>warning</b>            |
| log_location (on Linux only)                                     | Specifies the location of the PEM worker log file.  | On Linux, /var/log/pem/worker.log. On Windows, Logs & errors will be reported in the Application event log.          |
| agent_log_location (on Linux only)                               | Specifies the location of the PEM agent log file.   | On Linux, /var/log/pem/agent.log. On Windows, Logs & errors will be reported in the Application event log.           |
| ShortWait (on Windows) or short_wait (on Linux)                  | The minimum length of time (in seconds) that the PEM agent will wait before checking which probes are next in the queue (waiting to run).   | By default, 10 seconds.  |
| LongWait (on Windows) or long_wait (on Linux)                    | The maximum length of time (in seconds) that the PEM agent will wait before attempting to connect to the PEM server if an initial connection attempt fails.   | By default, 30 seconds.  |
| AlertThreads (on Windows) or alert_threads (on Linux)            | The number of alert threads to be spawned by the agent.   | Set to 1 for the agent that resides on the host of the PEM server; should be set to 0 for all other agents.          |
| EnableSMTP (on Windows) or enable_smtp (on Linux)                | When set to true for multiple PEM Agents (7.13 or lesser) and PEM backend database (9.4 or lesser) then it may send more duplicate emails. Whereas for PEM Agents (7.14 or higher) and PEM backend database (9.5 or higher) then it may send lesser duplicate emails. | By default, set to true for the agent that resides on the host of the PEM server; false for all other agents.        |
| EnableSNMP (on Windows) or enable_snmp (on Linux)                | When set to true for multiple PEM Agents (7.13 or lesser) and PEM backend database (9.4 or lesser) then it may send more duplicate traps. Whereas for PEM Agents (7.14 or higher) and PEM backend database (9.5 or higher) then it may send lesser duplicate traps.   | By default, set to true for the agent that resides on the host of the PEM server; false for all other agents.        |
| enable_nagios (on Linux only)                                    | When set to true, Nagios alerting is enabled.   | By default, set to true for the agent that resides on the host of the PEM server; false for all other agents.        |
| EnableWebhook (on Windows) or enable_webhook (on Linux)          | When set to true, Webhook alerting is enabled.  | By default, set to true for the agent that resides on the host of the PEM server; false for all other agents.        |
| MaxWebhookRetries (on Windows) or max_webhook_retries (on Linux) | Set maximum number of times pemAgent should retry to call webhooks on failure.  | Default 3.   |
| ConnectTimeout (on Windows) or connect_timeout (on Linux)        | The maximum length of time (in seconds, written as a decimal integer string) that the agent will wait for a connection.   | Not set by default. If set to 0, the agent will wait indefinitely.   |

| Parameter Name   | Description  | Value (if applicable)   |
|--|--|---|
| AllowServerRestart (on Windows) or allow_server_restart (on Linux)       | If set to TRUE, the agent can restart the database server that it monitors. Some PEM features may be enabled/disabled, depending on the value of this parameter.   | By default, set to TRUE.  |
| MaxConnections (on Windows) or max_connections (on Linux)                | The maximum number of probe connections used by the connection throttler.  | By default, set to 0 (an unlimited number of connections).  |
| ConnectionLifetime (on Windows) or connection_lifetime (on Linux)        | Use ConnectionLifetime (or connection_lifetime) to specify the minimum number of seconds an open but idle connection is retained. This parameter is ignored if the value specified in MaxConnections is reached and a new connection (to a different database) is required to satisfy a waiting request. | By default, set to 0 (a connection is dropped when the connection is idle after the agent's processing loop completes a cycle in which the connection has not been used).   |
| HeartbeatConnection (on Windows) or heartbeat_connection (on Linux)      | When set to TRUE, a dedicated connection used for sending the heartbeats.  | By default, set to FALSE.   |
| AllowBatchProbes (on Windows) or allow_batch_probes (on Linux)           | If set to TRUE, the user will be able to create batch probes using custom probes feature.  | By default, set to FALSE.   |
| BatchScriptDir (on Windows) or batch_script_dir (on Linux)               | Provide the path where script file (for alerting) will be stored.  | On Windows, C:\Users\user_name\AppData\LocalTemp. On Linux, set to /tmp.  |
| AllowBatchJobSteps (on Windows) or batch_script_user                     | Provide the username who will run the script.  | On Windows, set to TRUE and restart PEM Agent. Entries located in HKEY_LOCAL_MACHINE\Software\EnterpriseDB\PEM\agent. On Linux, Restart the agent after modifying the file. If you do not specify a user, or the specified user does not exist, then the script will not be executed. |
| ConnectionCustomSetup (on Windows) or connection_custom_setup (on Linux) | Use this parameter to provide SQL code that will be invoked each time a new connection with the monitored server is established.   | By default, no value is provided.   |
| ca_file (Linux only)   | Provide the path where the CA certificate resides.   | By default, /opt/PEM/agent/share/certs/ca-bundle.crt  |
| WebhookSSLKey (on Windows) or webhook_ssl_key (on Linux)                 | The complete path to the webhook's SSL client key file.  |   |
| WebhookSSLCrt (on Windows) or webhook_ssl.crt (on Linux)                 | The complete path to the webhook's SSL client certificate file.  |   |
| WebhookSSLCaCrt (on Windows) or webhook_ssl_ca.crt (on Linux)            | The complete path to the webhook's SSL ca certificate file.  |   |
| WebhookSSLCrl (on Windows) or webhook_ssl_crl (on Linux)                 | The complete path of the CRL file to validate webhook server certificate.  |   |
| AllowInsecureWebhooks (on Windows) or allow_insecure_webhooks (on Linux) | When set to true, allow webhooks to call with insecure flag.   | false   |

## Note

If you add or remove any of the parameter in the `agent.cfg` file then agent must be restarted to apply the changes.

## 34.2.7 PEM Agent Self Registration

Each PEM agent must be **registered** with the PEM server. The registration process provides the PEM server with the information it needs to communicate with the agent. The PEM agent graphical installer supports agent self-registration, but you can use the **pemworker** utility to register the agent if you skip PEM agent registration during a graphical installation or use an RPM package to install a PEM agent.

The RPM installer places the PEM worker utility in the `/usr/edb/pem/agent/bin` directory. Use the following commands to register an agent:

- **On Linux:** `pemworker --register-agent [register-options]`
- **On Windows:** `pemworker.exe REGISTER [register-options]`

The following information is required when registering an agent with the PEM Server; you will be prompted for information if it is not provided on the command line:

| Parameters               | Command-line options                               | Optional | Description  | Default Value   |
|--------------------------|--|----------|--|---|
| PEM Database Server Host | <code>--pem-server &lt;hostname/address&gt;</code> | No       | Address/Host name of the PEM database server   |   |
| PEM Admin User           | <code>--pem-user &lt;username&gt;</code>           | No       | <b>PEM Admin User</b> to connect to the PEM database server.   |   |
| PEM Database Server Port | <code>--pem-port &lt;port number&gt;</code>        | Yes      | Port on which PEM database server is running.  | 5432  |
| Agent Certificate Path   | <code>--cert-path &lt;certificate path&gt;</code>  | Yes      | Path, where certificates need to be created.   | On Linux, " <code>~/pem</code> " On Windows, " <code>%APPDATA%/pem</code> " |
| Agent Display Name       | <code>--display-name &lt;agent_name&gt;</code>     | Yes      | Display name of the PEM Agent.   | System hostname   |
| Agent Group              | <code>--group &lt;group_name&gt;</code>            | Yes      | The name of the group in which the agent will be displayed.  |   |
| Agent Team               | <code>--team &lt;team_name&gt;</code>              | Yes      | The name of the group role that may access the PEM Agent.  |   |
| Agent Owner              | <code>--owner &lt;owner_name&gt;</code>            | Yes      | The name of the owner of the PEM Agent.  |   |
| Force registration       | <code>--force-registration</code>                  | Yes      | Forcefully registers the agent to the PEM server with the arguments provided. It can be used to override the existing agent configuration.   |   |
| Enable Heartbeat         | <code>--enable-heartbeat-connection</code>         | Yes      | Agent to use dedicated connection to update the heartbeat.   | false   |
| Agent User               | <code>--pem-agent-user</code>                      | Yes      | Use this user to connect the PEM database server. Specify, it when you would like to use a connection pooler between PEM Agent and PEM database server. It will generate the SSL Certificates, which will used by the pemworker to connect to the PEM database server instead, for this user instead of the default agent user.<br><br><b>NOTE:</b> Specified user must be a member of 'pem_agent' role. |   |



**NOTE:** You can use the `PEM_SERVER_PASSWORD` environment variable to set the password of the `PEM Admin User`. If the `PEM_SERVER_PASSWORD` is not set, the server will use the `PGPASSWORD` or `pgpass file` when connecting to the `PEM Database Server`.

Example:

```

root@localhost:/usr/edb/pem/agent/etc
File Edit View Search Terminal Help
[root@localhost yum.repos.d]# cd /usr/edb/pem/agent/bin
[root@localhost bin]# ls
pemagent pemworker pkgLauncher
[root@localhost bin]# export PEM_SERVER_PASSWORD=postgres
[root@localhost bin]# ./pemworker --register-agent
Postgres Enterprise Manager Server Hostname: 192.168.1.16
Postgres Enterprise Manager Server Username: postgres
Postgres Enterprise Manager Server Port: 5432
Postgres Enterprise Manager Agent registered successfully!

```

Refer the [PEM Worker Usage Guide](#).

## 34.2.8 Register/Unregister database server using PEM Agent

You can use the `pemworker` utility to register a database server for monitoring. The RPM installer places the utility in the `/usr/edb/pem/agent/bin` directory. Use the following commands to register a server:

- **On Linux:** `pemworker --register-server [register-server-options]`
- **On Windows:** `pemworker.exe REGISTER-SERVER [register-server-options]`

Use the parameters in the table that follow to provide connection information for a Postgres database server that you wish to register for monitoring by the PEM Server. Please note that the `pg_hba.conf` file on the database server must be configured to allow connections from the PEM server.

Properties that begin with `--asb` (agent server binding) define the binding for an agent that does not reside on the same host as the database server. These properties are optional if you have a PEM agent installed on the host of the database server. You will be prompted for required information if you do not include it on the command line.

| Parameters          | Command-line options                                 | Optional | Description  | Default Value |
|---------------------|--|----------|--|---------------|
| PEM Admin User      | <code>--pem-user &lt;username&gt;</code>             | No       | The name of the <code>PEM Admin User</code> that will connect to the monitored server.                                 |               |
| Server Host         | <code>--server-addr &lt;host name/address&gt;</code> | No       | Host name/address of the monitored server.   |               |
| Server Port         | <code>--server-port &lt;port&gt;</code>              | No       | Port on which database server is running.  |               |
| Server Database     | <code>--server-database &lt;name&gt;</code>          | No       | The database to which PEM will connect.  |               |
| Server User         | <code>--server-user &lt;name&gt;</code>              | No       | The database user role that will be used by the agent for monitoring purposes.   |               |
| Server Service Name | <code>--server-service-name &lt;name&gt;</code>      | Yes      | Name of the system level service, which controls the operations like start, stop, restart, reload, etc. of the server. |               |

| Parameters                                    | Command-line options              | Optional | Description  | Default Value  |
|---|-----------------------------------|----------|--|--|
| <b>Remote Monitoring?</b>                     | --remote-monitoring <yes/no>      | No       | <b>no</b> if the monitored server resides on the same machine as the bound PEM agent, <b>yes</b> if the agent is on another host.  | no   |
| <b>- EDB Failover Manager</b>                 |                                   |          |  |  |
| Cluster Name                                  | --efm-cluster-name <name>         | Yes      | Name of EDB Failover Manager Cluster associated with this server.  |  |
| <b>EDB Failover manager Installation Path</b> | --efm-install-path <path>         | Yes      | Installation path of EDB Failover Manager associated with this server.   |  |
| <b>Server Display Name</b>                    | --display-name <server_name>      | Yes      | Display name of the registered server.   | System hostname  |
| <b>Host Name</b>                              | --asb-host-name <name_of_host>    | Yes      | The name of the host to which the agent is connecting.   | The value specified by the --server-addr property.     |
| <b>Host Port</b>                              | --asb-host-port <port_number>     | Yes      | The port number that the agent will use when connecting to the database.   | The value specified by the --server-port property.     |
| <b>Host DB</b>                                | --asb-host-db <database_name>     | Yes      | The name of the database to which the agent will connect.  | The value specified by the --server-database property. |
| <b>Host User Name</b>                         | --asb-host-user <database_user>   | Yes      | The database user name that the agent will supply when authenticating with the database.   | The value specified by the --server-user property.     |
| <b>SSL Mode</b>                               | --asb-ssl-mode <certificate path> | Yes      | Type of SSL authentication that will be used for connections. Supported values include <b>prefer</b> , <b>require</b> , <b>disable</b> , <b>verify-CA</b> , <b>verify-full</b> . | prefer   |
| <b>Server Group</b>                           | --group <group_name>              | Yes      | Specify the name of the server group in which the server will be displayed.  |  |
| <b>Server Team</b>                            | --team <team_name>                | Yes      | Specify the name of the group role that will be allowed to access the server.  |  |
| <b>Owner</b>                                  | --owner <owner_name>              | Yes      | Specify the name of the role that will own the monitored server.   |  |

Use the **PEM\_MONITORED\_SERVER\_PASSWORD** environment variable to set the password of the user of the database server which is to be registered. When registering the database server, the pemworker utility will bind the server to the **PEM Agent** associated with the pemworker utility. The PEM server will use the specified user name ( **Server User** ) and password specified in the **PEM\_MONITORED\_SERVER\_PASSWORD** environment variable when monitoring the database server.

Use the **PEM\_SERVER\_PASSWORD** environment variable to provide the password of the user of the PEM database server. If the **PEM\_SERVER\_PASSWORD** is not set, the server will use the **PGPASSWORD** or **pgpass file** when connecting to the **PEM Database Server**.

## To unregister a database server

You can use the pemworker utility to unregister a server:

- **On Linux:** pemworker --unregister-server [unregister-server-options]
- **On Windows:** pemworker.exe UNREGISTER-SERVER [unregister-server-options]

Include the following information when unregistering a database server from the **PEM Server**; you will be prompted for required information if you do not include it on the command line:

| Parameters     | Command-line options              | Optional | Description  | Default Value |
|----------------|-----------------------------------|----------|--|---------------|
| PEM Admin User | --pem-user <username>             | No       | PEM Admin User to connect the PEM database server. |               |
| Server Host    | --server-addr <host name/address> | No       | Host name/address of the database server.          |               |
| Server Port    | --server-port <port>              | No       | Port on which database server is running.          |               |

The command will unregister the server from the PEM Server for the specified combination of Server Host and Server Port, which is being monitored by the PEM Agent.

For more information, refer the [PEM Worker Usage Guide](#).

## 34.3 The PEM Client

The Postgres Enterprise Manager client provides a powerful and intuitive user interface that you can use to manage Advanced Server and PostgreSQL databases. The client interface is easily customized, and will preserve your preferences between sessions. Client features include:

- auto-detection and support for objects discovered at run-time
- a live SQL query tool with direct data editing
- support for administrative queries
- a syntax-highlighting SQL editor
- powerful graphical management dialogs and tools for common tasks
- a responsive, context-sensitive behavior
- supportive error messages
- helpful hints
- online help and information for dialogs and tools.

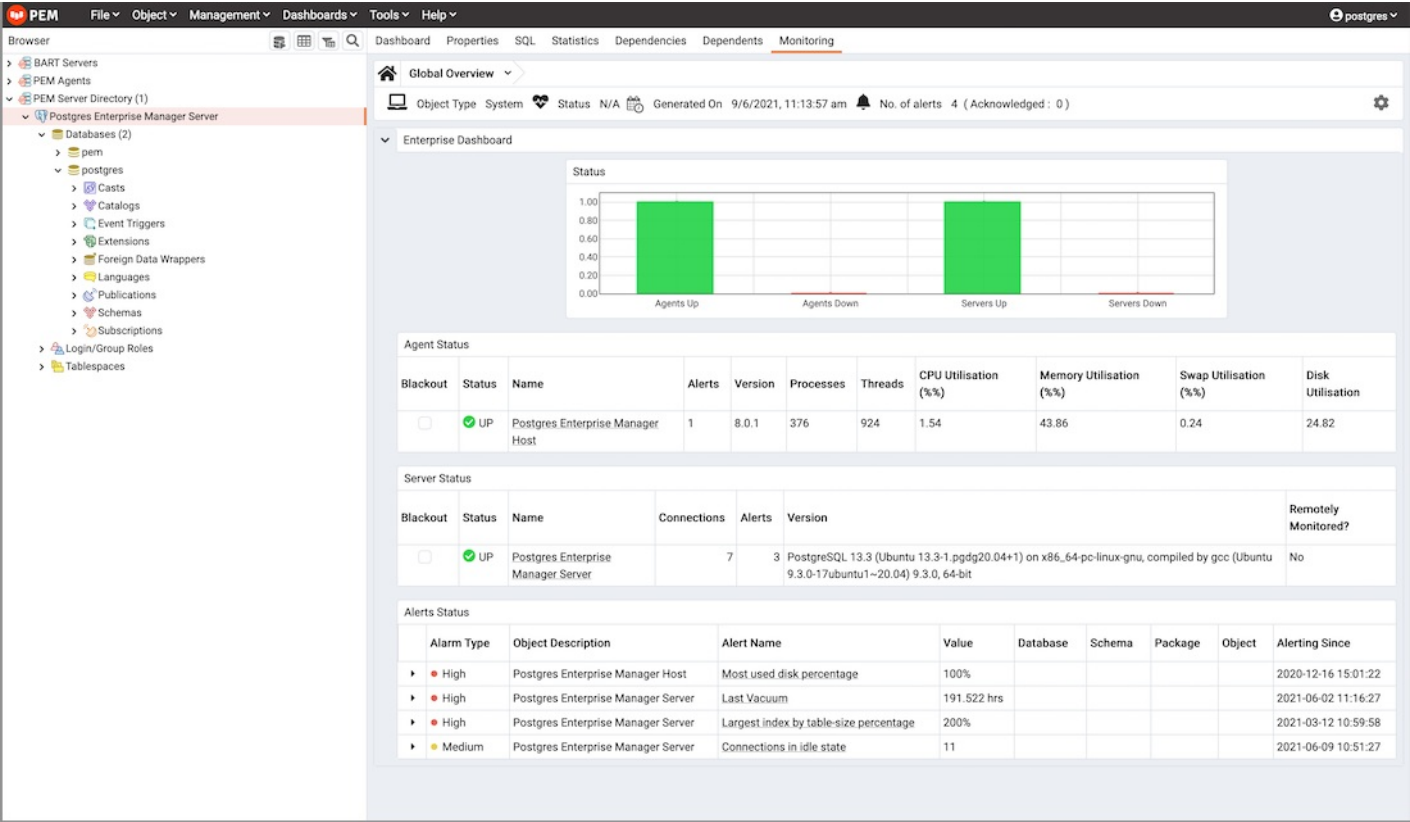
The PEM client features a highly-customizable display that features drag-and-drop panels that you can arrange to make the best use of your desktop environment. The application is installed during the PEM server installation; use your browser of choice to connect to the client.

The client tree control (the *Browser*) provides an elegant overview of the managed servers, and the objects that reside on each server. Right-click on a node within the tree control to access context-sensitive menus that provide quick access to management tasks for the selected object. The tabbed browser window provide quick access to statistical information about each object in the tree control, tools and utilities, and extended PEM features. The client opens an additional feature tab each time you access the extended functionality offered by PEM; you can open, close, and re-arrange tabs as needed.

You can search for objects in the database using the [Search objects](#)

### 34.3.1 PEM Main Browser Window

The PEM client features a menu bar and a window divided into two panes: the Browser tree control in the left pane, and a tabbed browser in the right pane.



Menus displayed across the top of the browser window provide quick, context-sensitive access to PEM features and functionality.

The PEM Client Object Browser

The **Browser** tree control provides access to information and management options for the database objects that reside on each server. The tree control expands to display a hierarchical view of the servers and objects that are monitored by the PEM server. You can use context menu options (accessed by right-clicking on nodes of the tree control) to create new objects, and modify and delete existing objects if your role holds the required privileges.

Expand nodes in the tree control to display a hierarchical view of the database objects that reside on a selected server:

- Use the plus sign (+) to the left of a node to expand a segment of the tree control.
- Click the minus sign (-) to the left of a node to close that node.

Right-click on a node of the tree control to access a context-sensitive menu and perform common tasks. Context menu options may include one or more of the following selections:

| Option                  | Action  |
|-------------------------|---|
| Add named restore point | Click to create and enter the name of a restore point.  |
| Backup...               | Click to open the Backup... dialog to backup database objects.  |
| Backup Globals...       | Click to open the Backup Globals... dialog to backup cluster objects.   |
| Backup Server...        | Click to open the Backup Server... dialog to backup a server.   |
| Connect Server          | Click to establish a connection with the selected server.   |
| Create                  | Click to access a context menu that provides context-sensitive selections.Your selection opens a Create dialog for creating a new object. |
| CREATE Script           | Click to open the Query tool to edit or view the CREATE script.   |
| CREATE Script           | Click to open the Query tool to edit or view the CREATE script.   |

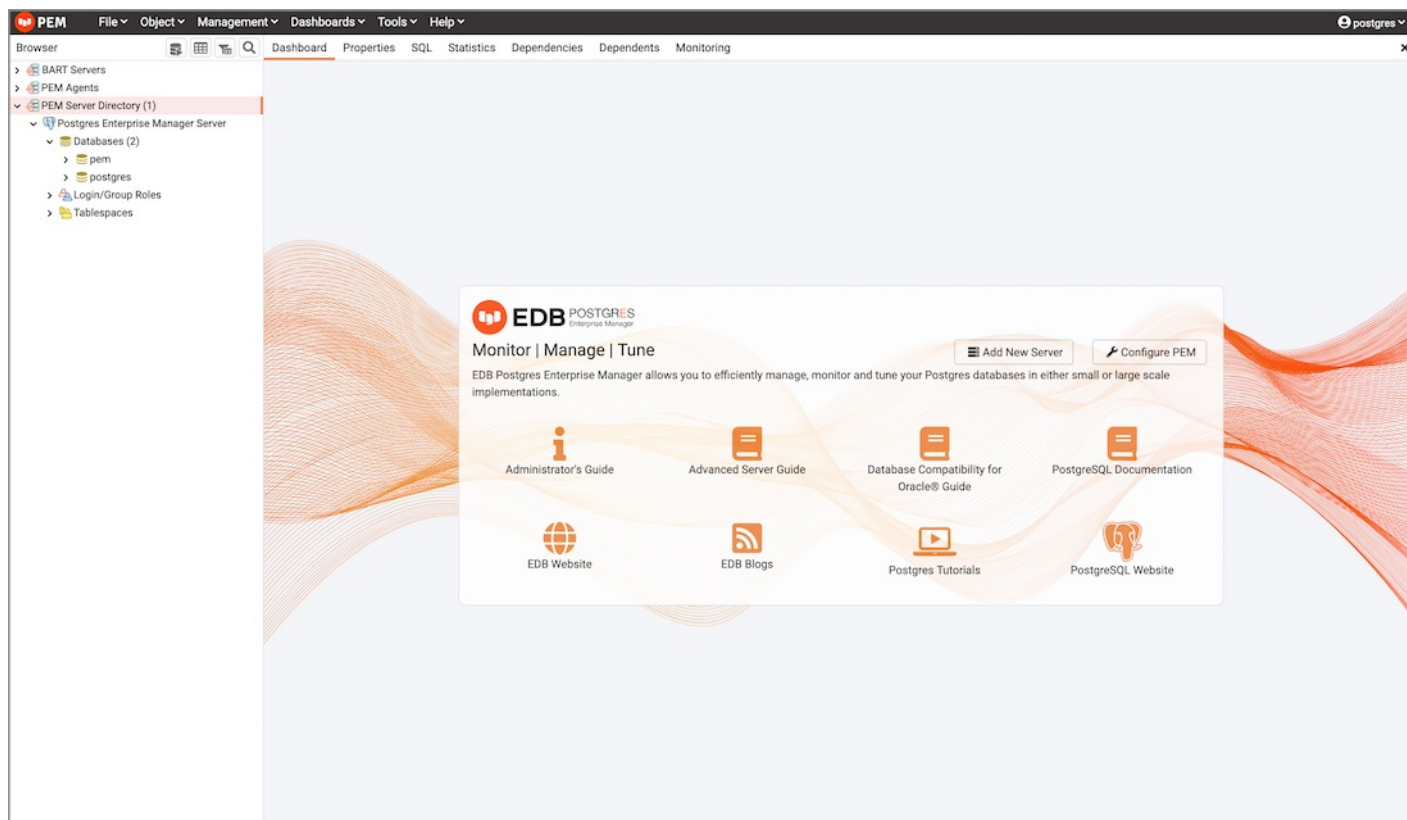
| Option                  | Action  |
|-------------------------|---|
| Dashboards              | Click through for quick access to PEM dashboards.   |
| Delete/Drop             | Click to delete the currently selected object from the server.  |
| Disconnect Database...  | Click to terminate a database connection.   |
| Disconnect Server...    | Click to refresh the currently selected object.   |
| Drop Cascade            | Click to delete the currently selected object and all dependent objects from the server.  |
| Debugging               | Click to access the <a href="#">Debugger</a> tool.  |
| Grant Wizard            | Click to access the <a href="#">Grant Wizard</a> tool.  |
| Maintenance...          | Click to open the <a href="#">Maintenance...</a> dialog to VACUUM, ANALYZE, REINDEX, or CLUSTER.  |
| Management              | Click to access management tasks that are relevant to the node.   |
| Properties...           | Click to review or modify the currently selected object's properties.   |
| Refresh...              | Click to refresh the currently selected object.   |
| Reload Configuration... | Click to update configuration files without restarting the server.  |
| Restore...              | Click to access the <a href="#">Restore</a> dialog to restore database files from a backup.   |
| View Data               | Use the <a href="#">View Data</a> option to access the data stored in a selected table with the <a href="#">Data Output</a> tab of the <a href="#">Query tool</a> . |

The context-sensitive menus associated with [Tables](#) and nested [Table](#) nodes provides additional display options:

| Option                | Action  |
|-----------------------|---|
| Import/Export...      | Click open the [Import/Export...](#import_export_data) dialog to import data to or export data from the selected table. |
| Reset Statistics      | Click to reset statistics for the selected table.   |
| Scripts               | Click to open the <a href="#">Query tool</a> to edit or view the selected script from the flyout menu.                  |
| Truncate              | Click to remove all rows from a table.  |
| Truncate Cascade      | Click to remove all rows from a table and its child tables.   |
| View First 100 Rows   | Click to access <a href="#">the data grid</a> that displays the first 100 rows of the selected table.                   |
| View Last 100 Rows    | Click to access <a href="#">the data grid</a> that displays the last 100 rows of the selected table.                    |
| View All Rows         | Click to access <a href="#">the data grid</a> that displays all rows of the selected table.                             |
| View Filtered Rows... | Click to access the <a href="#">Data Filter</a> popup to apply a filter to a set of data.                               |

## The PEM Tabbed Browser Window

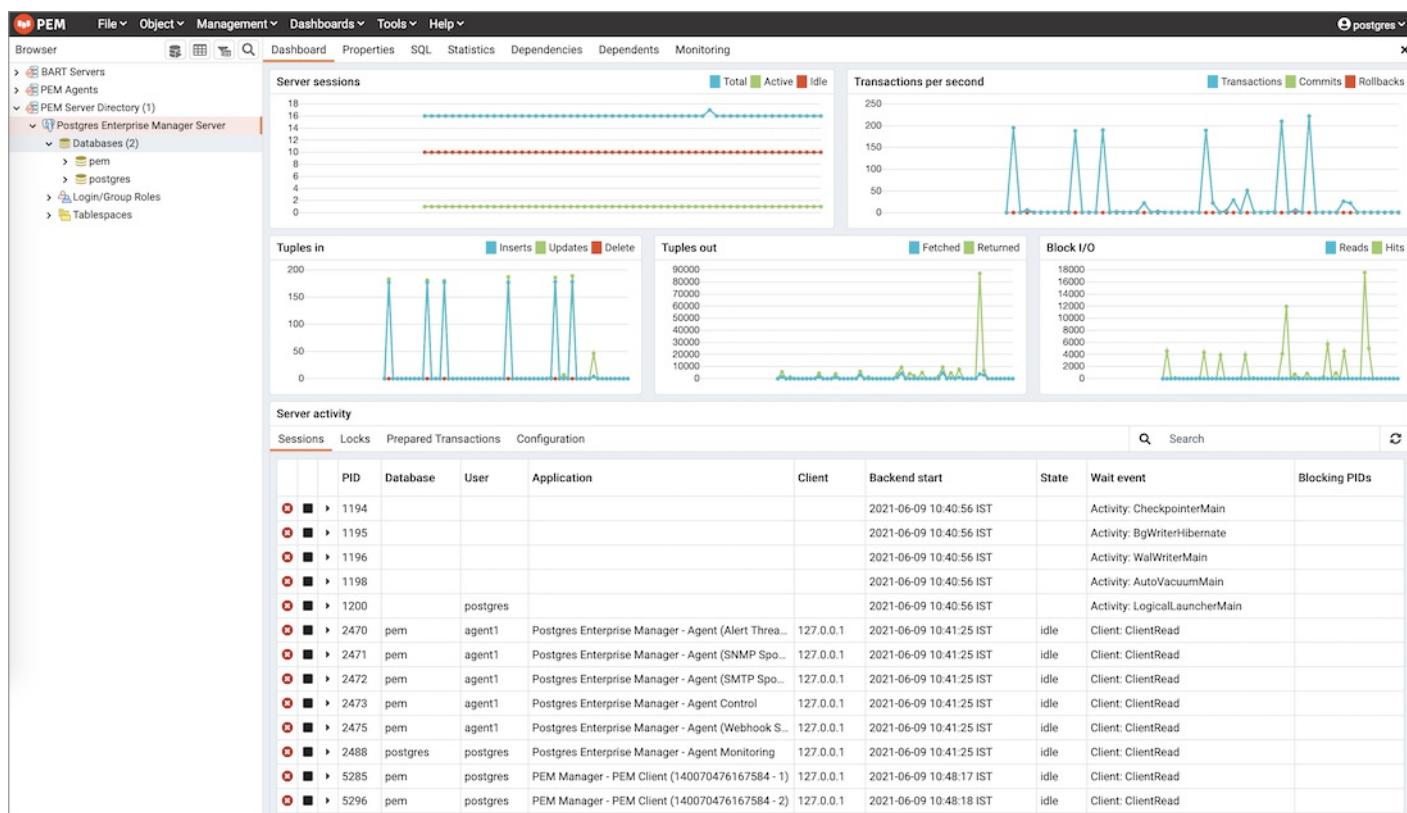
The main panel of the PEM client contains a collection of tabs that display information about the object currently selected in the tree control.



The **Dashboard** tab is context-sensitive; when you navigate to the **Dashboard** tab from a server group or the **PEM Agents** node, the EDB Postgres **Welcome** window opens, allowing you to:

- Click the **Add New Server** icon to open the **Create - Server dialog** to define a connection to a server.
- Click the **Configure PEM** icon to open the **Server Configuration dialog** and modify server parameters.
- Click the **Getting Started** icon to open a new tab, displaying the PEM Getting Started section at the EnterpriseDB website.
- Click the **EDB Website** icon to navigate to the home page of the EnterpriseDB website. The EnterpriseDB website features news about upcoming events and other projects.
- Click the **PostgreSQL Website** icon to navigate to the PostgreSQL project website. The PostgreSQL site features news about recent releases and other project information.
- Click the **EDB Blogs** icon to navigate to the EDB Blog page, where you can review the most-recent employee posts to Postgres related blogs.

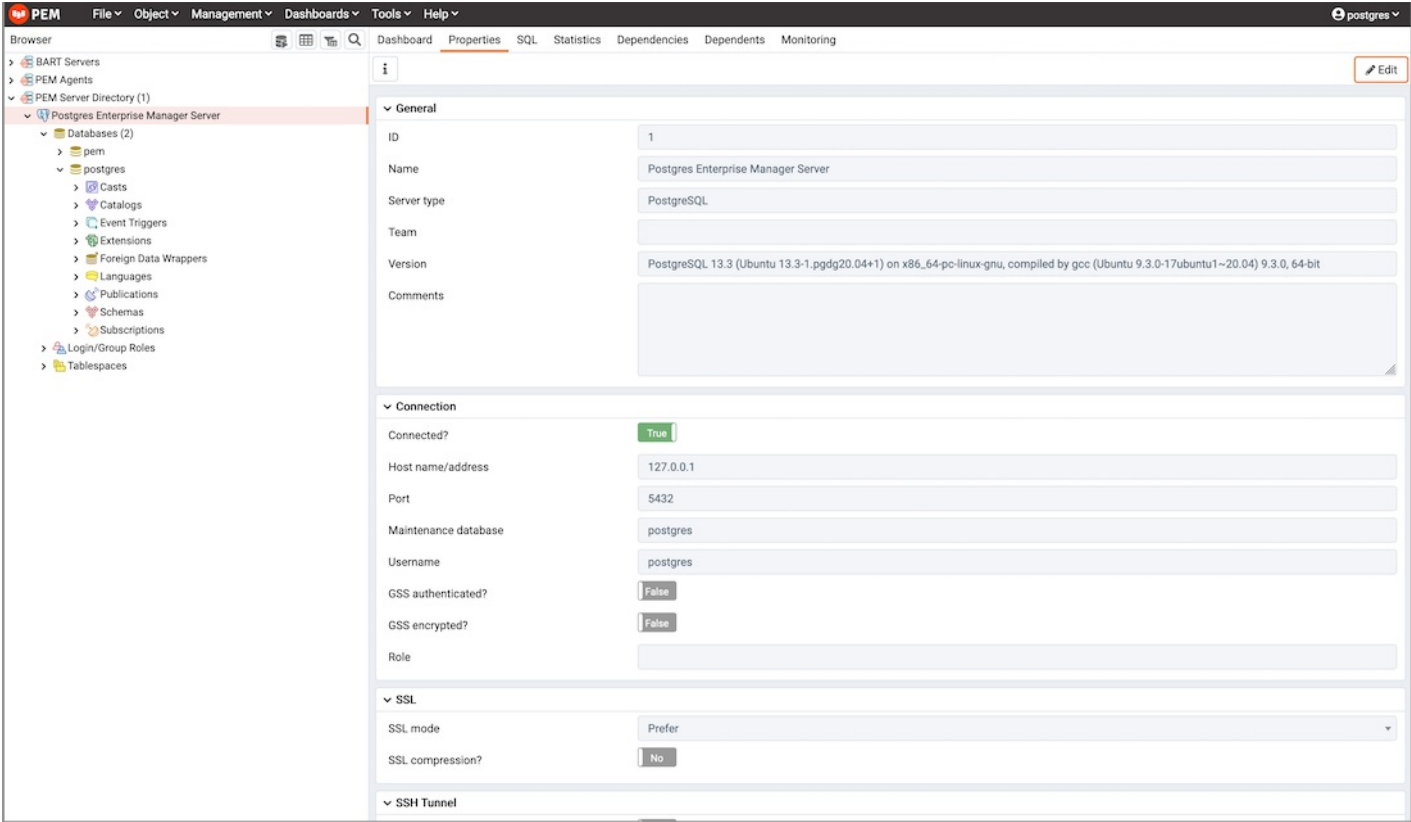
Highlight the name of an agent or server and navigate to the **Dashboard** tab to review session or server activity for the currently selected object.



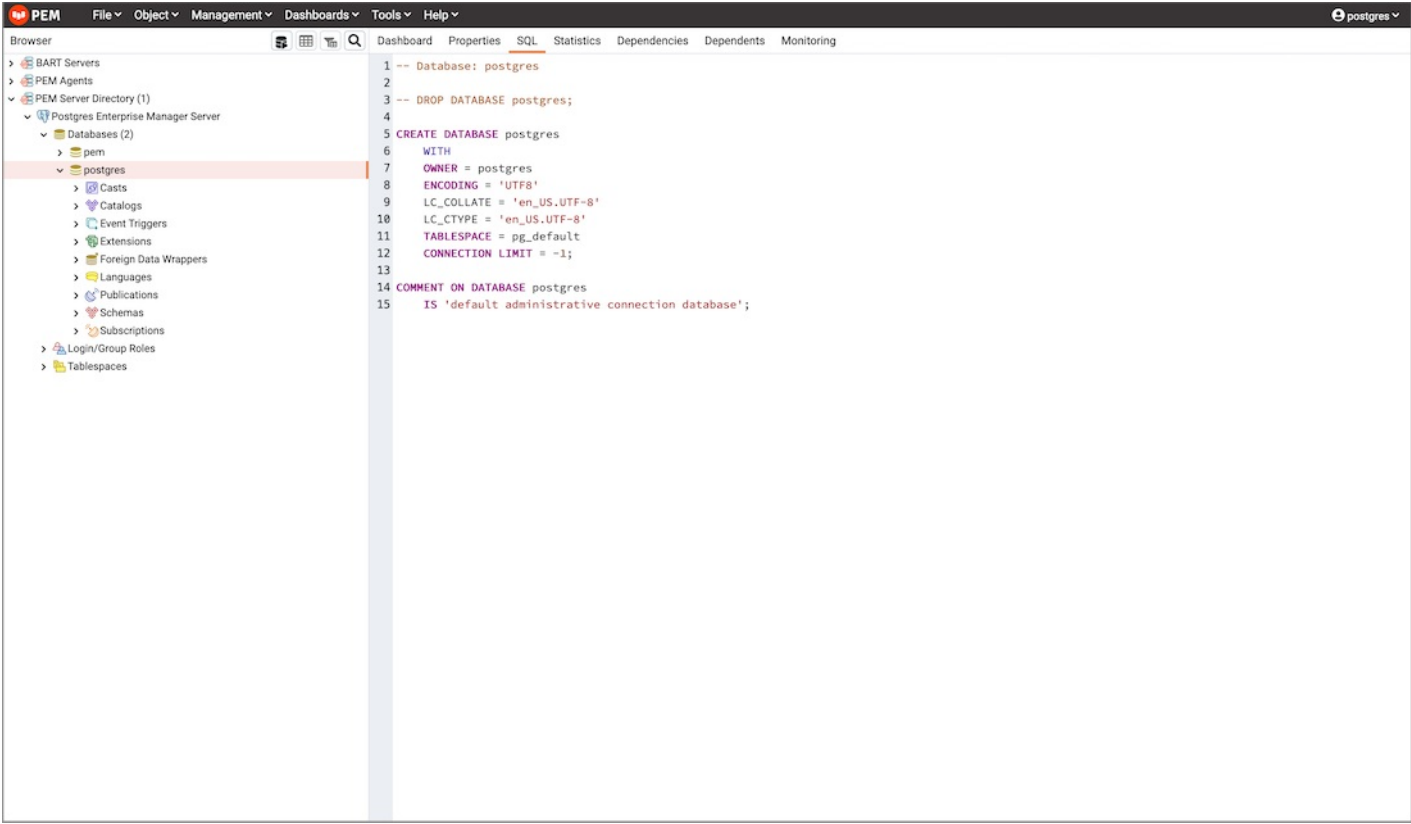
When opened from the name of an agent or server, the **Dashboard** tab provides a graphical analysis of usage statistics:

- The **Server sessions** or **Database sessions** graph displays the interactions with the server or database.
- The **Transactions per second** graph displays the commits, rollbacks, and total transactions per second that are taking place on the server or database.
- The **Tuples In** graph displays the number of tuples inserted, updated, and deleted on the server or database.
- The **Tuples out** graph displays the number of tuples fetched and returned from the server or database.
- The **Block I/O** graph displays the number of blocks read from the filesystem or fetched from the buffer cache (but not the operating system's file system cache) for the server or database.
- The **Server activity** tabbed panel displays tables that contain session information, session locks, prepared transactions and configuration.





Navigate to the **Properties** tab to review the properties of the item currently highlighted in the tree control.



The **SQL** tab displays the SQL code used to generate the object currently selected in the Browser tree control.



PEM

FileObjectManagementDashboardsToolsHelp

Browser

DashboardPropertiesSQLStatisticsDependenciesDependentsMonitoring

BART Servers

PEM Agents

PEM Server Directory (1)

Postgres Enterprise Manager Server

Databases (2)

pem

postgres

Casts

Catalogs

Event Triggers

Extensions

Foreign Data Wrappers

Languages

Publications

Schemas

Subscriptions

Login/Group Roles

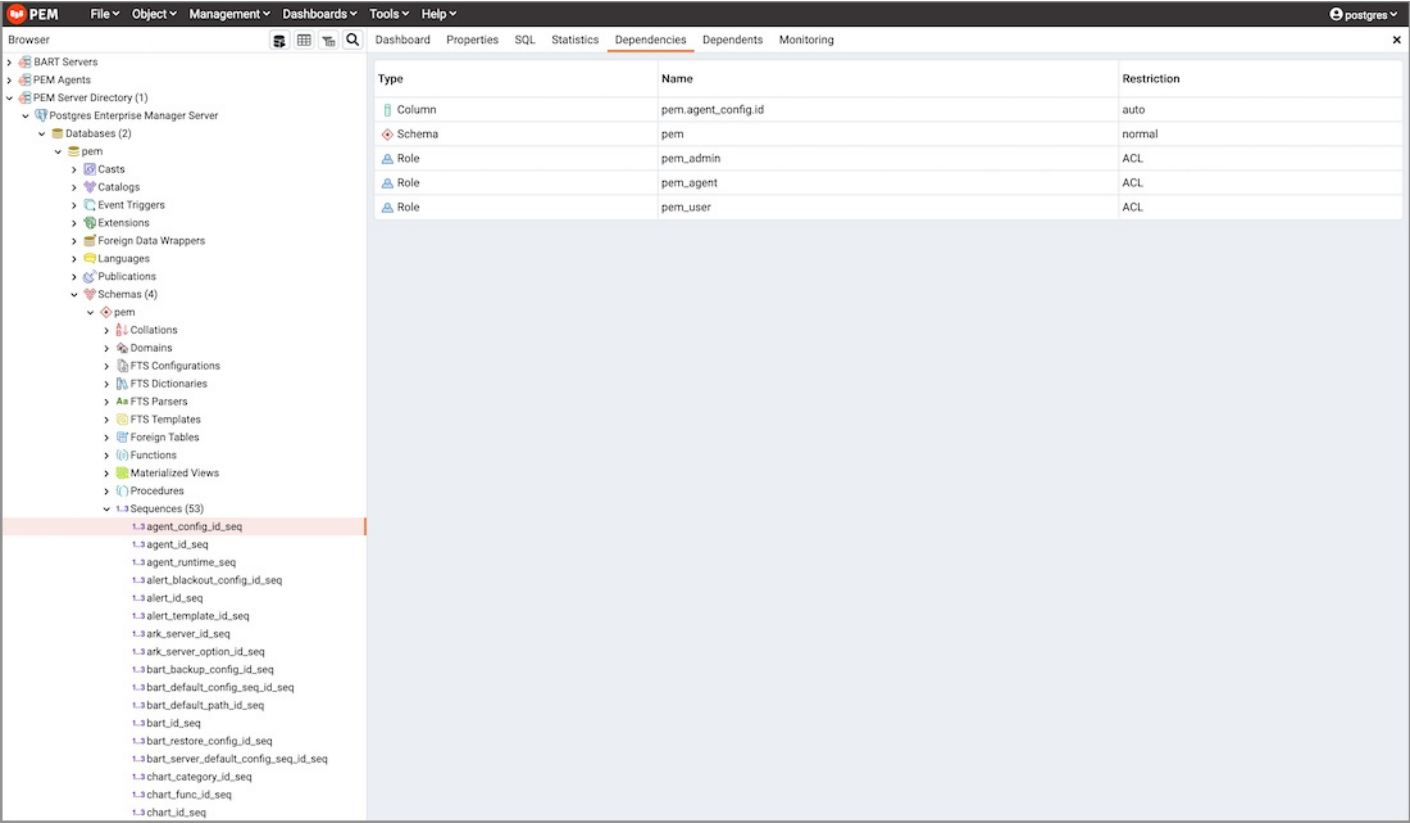
Tablespaces

| PID  | User     | Database | Backend start                    | Client             | Application                            | Wait event type | Wait event name     | Query                                     | Query start     |
|------|----------|----------|----------------------------------|--------------------|--|-----------------|---------------------|---|-----------------|
| 1196 |          |          | 2021-06-09 10:40:56.209773+05:30 |                    |  | Activity        | WaiWriterMain       |   |                 |
| 9032 | postgres | postgres | 2021-06-09 11:27:02.646451+05:30 | 127.0.0.1/32:59168 | Postgres Enterprise Manager - Agent... | Client          | ClientRead          | SELECT 1                                  | 2021-06-09 11:4 |
| 2475 | agent1   | pem      | 2021-06-09 10:41:25.832501+05:30 | 127.0.0.1/32:54872 | Postgres Enterprise Manager - Agent... | Client          | ClientRead          | END                                       | 2021-06-09 11:4 |
| 5285 | postgres | pem      | 2021-06-09 10:48:17.409341+05:30 | 127.0.0.1/32:56142 | PEM Manager - PEM Client (1400704...   | Client          | ClientRead          | SELECT count(*) FROM pg_catalog p...      | 2021-06-09 10:4 |
| 2470 | agent1   | pem      | 2021-06-09 10:41:25.751473+05:30 | 127.0.0.1/32:54864 | Postgres Enterprise Manager - Agent... | Client          | ClientRead          | SELECT pem.process_one_alert()            | 2021-06-09 11:4 |
| 6130 | postgres | pem      | 2021-06-09 10:49:38.251428+05:30 | 127.0.0.1/32:56332 | PEM Client - DB:pem                    | Client          | ClientRead          | SELECT ref.relname AS refname, d2.r...    | 2021-06-09 11:1 |
| 2473 | agent1   | pem      | 2021-06-09 10:41:25.820732+05:30 | 127.0.0.1/32:54870 | Postgres Enterprise Manager - Agent... | Client          | ClientRead          | SELECT heartbeat_interval, active, arr... | 2021-06-09 11:4 |
| 5296 | postgres | pem      | 2021-06-09 10:48:18.392134+05:30 | 127.0.0.1/32:56156 | PEM Manager - PEM Client (1400704...   | Client          | ClientRead          | WITH chart_cfg AS ( SELECT c.id AS ...    | 2021-06-09 11:4 |
| 2472 | agent1   | pem      | 2021-06-09 10:41:25.809603+05:30 | 127.0.0.1/32:54868 | Postgres Enterprise Manager - Agent... | Client          | ClientRead          | END                                       | 2021-06-09 11:4 |
| 6112 | postgres | postgres | 2021-06-09 10:49:37.94856+05:30  | 127.0.0.1/32:56324 | PEM Client - DB:postgres               |                 |                     | SELECT pid AS "PID", username AS "U...    | 2021-06-09 11:4 |
| 5307 | postgres | pem      | 2021-06-09 10:48:18.587637+05:30 | 127.0.0.1/32:56158 | PEM Manager - PEM Client (1400704...   | Client          | ClientRead          | SELECT CASE WHEN ((SELECT count...        | 2021-06-09 11:4 |
| 1194 |          |          | 2021-06-09 10:40:56.202133+05:30 |                    |  | Activity        | CheckpointMain      |   |                 |
| 1195 |          |          | 2021-06-09 10:40:56.216782+05:30 |                    |  | Activity        | BgWriterHiberate    |   |                 |
| 1200 | postgres |          | 2021-06-09 10:40:56.231055+05:30 |                    |  | Activity        | LogicalLauncherMain |   |                 |
| 2471 | agent1   | pem      | 2021-06-09 10:41:25.798144+05:30 | 127.0.0.1/32:54866 | Postgres Enterprise Manager - Agent... | Client          | ClientRead          | END                                       | 2021-06-09 11:4 |
| 1198 |          |          | 2021-06-09 10:40:56.207252+05:30 |                    |  | Activity        | AutoVacuumMain      |   |                 |

The **Statistics** tab displays the statistics gathered for each object on the tree control; the statistics displayed in the table vary by the type of object that is highlighted. Click a column heading to sort the table by the data displayed in the column; click again to reverse the sort order. The following table lists some of the statistics that may be displayed:

| Panel                 | Description  |
|-----------------------|--|
| PID                   | The process ID associated with the row.  |
| User                  | The name of the user that owns the object.   |
| Database              | displays the database name.  |
| Backends              | displays the number of current connections to the database.  |
| Backend start         | The start time of the backend process.   |
| Xact Committed        | displays the number of transactions committed to the database within the last week.                        |
| Xact Rolled Back      | displays the number of transactions rolled back within the last week.                                      |
| Blocks Read           | displays the number of blocks read from memory (in megabytes) within the last week.                        |
| Blocks Hit            | displays the number of blocks hit in the cache (in megabytes) within the last week.                        |
| Tuples Returned       | displays the number of tuples returned within the last week.   |
| Tuples Fetched        | displays the number of tuples fetched within the last week.  |
| Tuples Inserted       | displays the number of tuples inserted into the database within the last week.                             |
| Tuples Updated        | displays the number of tuples updated in the database within the last week.                                |
| Tuples Deleted        | displays the number of tuples deleted from the database within the last week.                              |
| Last statistics reset | displays the time of the last statistics reset for the database.   |
| Tablespace conflicts  | displays the number of queries canceled because of recovery conflict with dropped tablespaces in database. |
| Lock conflicts        | displays the number of queries canceled because of recovery conflict with locks in database.               |
| Snapshot conflicts    | displays the number of queries canceled because of recovery conflict with old snapshots in database.       |
| Bufferpin conflicts   | displays the number of queries canceled because of recovery conflict with pinned buffers in database.      |
| Temporary files       | displays the total number of temporary files, including those used by the statistics collector.            |

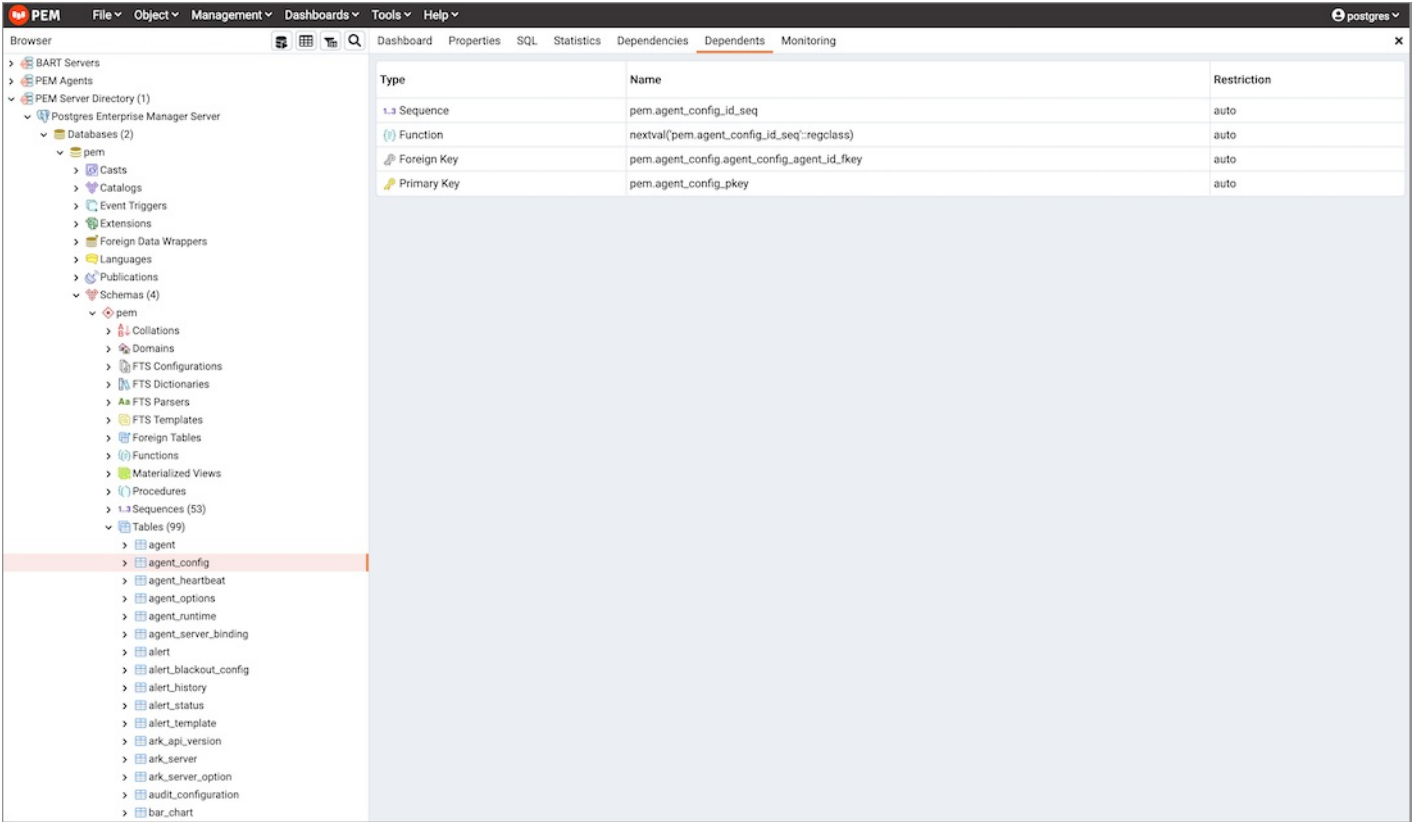
| Panel                   | Description  |
|-------------------------|--|
| Size of temporary files | displays the size of the temporary files.  |
| Deadlocks               | displays the number of queries canceled because of a recovery conflict with deadlocks in database. |
| Block read time         | displays the number of milliseconds required to read the blocks read.                              |
| Block write time        | displays the number of milliseconds required to write the blocks read.                             |
| Size                    | displays the size (in megabytes) of the selected database.   |



The **Dependencies** tab displays the objects on which the currently selected object depends. To ensure the integrity of the database structure, the server makes sure that you do not accidentally drop objects that other objects depend on; you must use DROP CASCADE to remove an object on which another object depends.

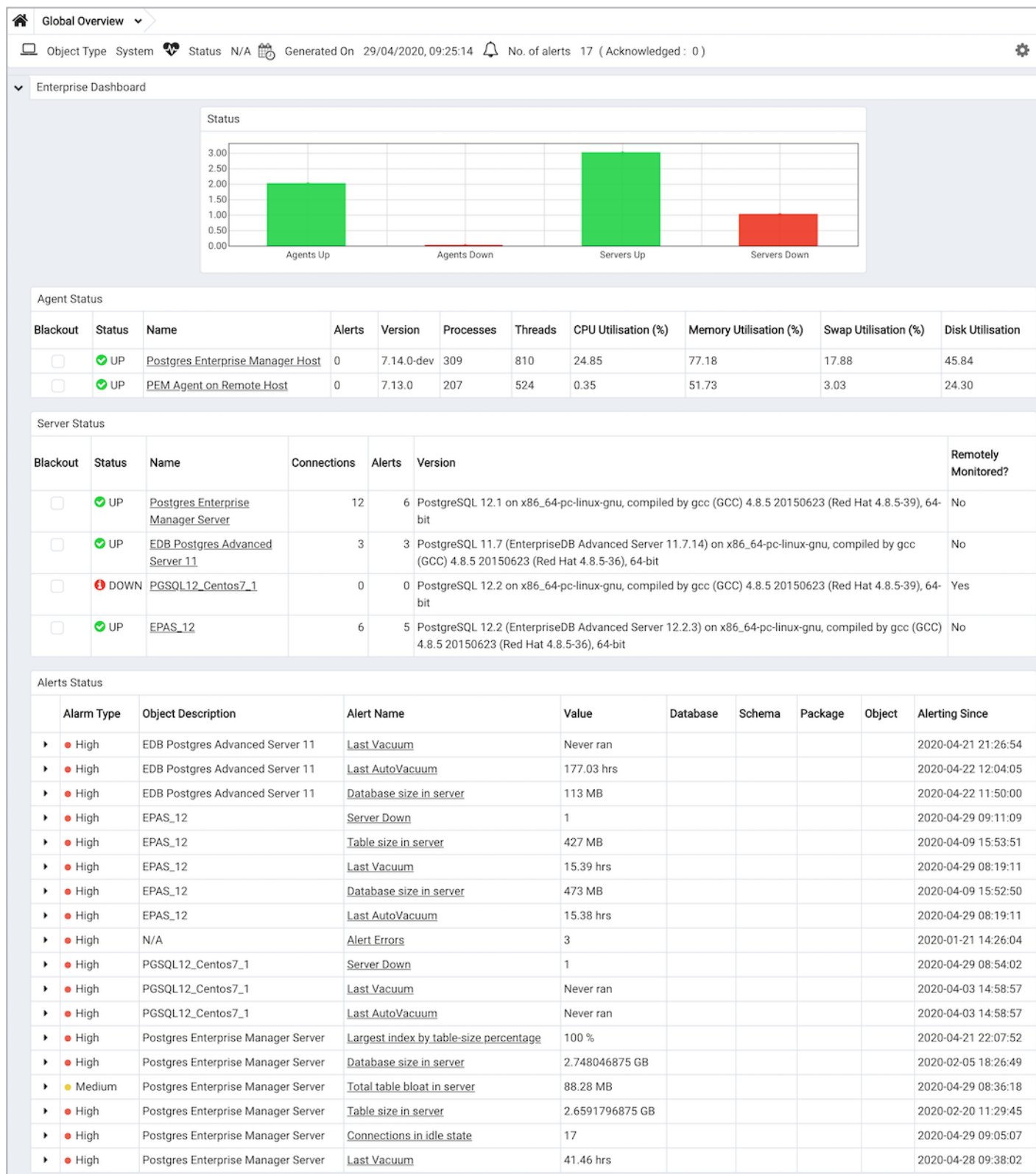
The **Dependencies** table displays:

- The **Type** field specifies the parent object type.
- The **Name** field specifies the identifying name of the parent object.
- The **Restriction** field describes the dependency relationship between the currently selected object and the parent.



The **Dependents** tab displays a table of objects that depend on the object currently selected in the Browser tree. A dependent object can be dropped without affecting the object currently selected in the Browser tree control.

- The **Type** field specifies the dependent object type.
- The **Name** field specifies the identifying name for the dependent object.
- The **Restriction** field describes the dependency relationship between the currently selected object and the parent.
- Navigate to the **Monitoring** tab to access information presented on **PEM dashboards**. Dashboards display statistical information about the objects monitored by the PEM server.



PEM will open additional tabs when you access PEM functionality through the **Management** or **Tools** dialogs. Right-click the current tab and select from a context menu that allows you to customize the display for your working style:


- Click **Remove Panel** to remove the currently selected panel.
- Click **Rename Panel** to rename the currently selected panel.
- Click **Detach Panel** to detach the currently selected panel, repositioning it for convenience.
- Click **Add Panel** and select any of the available options to be added to the panels.


The PEM client will preserve any adjustments when you exit the program; to reset the PEM client to its original format, select **Reset Layout** from the **File** menu.


## Using Chart, Graph and Table Controls


Use the icons in the upper-right corner of each graphic on a PEM Client dashboard to control, download, and customize the charts, graphs and tables displayed in the PEM client.




Use the **Refresh** icon  to display the most-recent content available from the PEM probes.

Select the **Download** icon  to download a .jpeg or .png image of the chart or graph. By default, the file will be in .jpeg format; to save the file as a .png, use the **Personalize** icon to modify the download format.

Select the **Fullscreen** icon  to expand the chart or graph to fill the main pane of the PEM client.

Select the **Personalize** icon  to modify the display properties of the chart or graph for your session only.

Use the **Information** icon  to access information about the chart or graph.

### Personalizing a Graphic

When you select the **Personalize** icon, the **Personalize chart configuration** dialog opens:

Use controls on the **Personalize chart configuration** dialog to modify the properties of the graphic:

- Use the **Auto Refresh** control to increase or decrease the number of seconds between refreshes.
- Use the **Auto Refresh** field to specify the number of seconds between updates of the data displayed in the table or chart.
- If applicable, use the **Download as** field to indicate if you would like a chart to be downloaded as a JPEG image or a PNG image.
- If applicable, use the **Colours** selectors to specify the display colors that will be used on a chart.
- If applicable, set the **Show Acknowledged Alerts** switch to **Yes** indicate that you would like the table to display alerts that you have acknowledged with a checkbox in the **Ack'ed** column. Set the field to **No** to indicate that the table should hide any acknowledged alerts. The switch acts as a toggle; acknowledged alerts are not purged from the table content until the time specified in the alert definition passes.

After personalizing the display properties, use the controls in the upper-right hand corner to apply your changes:

- Use the **Delete** icon to reset the properties of the graphic to their default settings; use the drop-down listbox to access a menu that allows you to apply the change to only this instance of the graphic, or to the same graphic when displayed on other dashboards.
- Use the **Save** icon to save your changes to the properties for the graphic; use the drop-down listbox to access a menu that allows you to apply the change to only this instance of the graphic, or to the same graphic when displayed on other dashboards.

- Click the X to close the dialog without changing the properties of the graphic.

### 34.3.2 Browser Toolbar

The browser toolbar provides shortcut buttons for frequently used features like View Data and the Query tool which are most frequently used in PEM. This toolbar is visible on the Browser panel. Buttons get enabled/disabled based on the selected browser node.

PEM File Object Management

Browser

- > BART Servers
- > PEM Agents
- ▼ PEM Server Directory (1)
  - ▼ Postgres Enterprise Manager Server
    - ▼ Databases (2)
      - > pem
      - ▼ postgres
        - > Casts
        - > Catalogs
        - > Event Triggers
        - > Extensions
        - > Foreign Data Wrappers
        - > Languages
        - > Publications
        - ▼ Schemas (1)
          - ▼ public
            - > Collations
            - > Domains
            - > FTS Configurations
            - > FTS Dictionaries
            - > FTS Parsers
            - > FTS Templates
            - > Foreign Tables
            - > Functions
            - > Materialized Views
            - > Procedures
            - > Sequences
            - > Tables
            - > Trigger Functions
            - > Types
            - > Views
          - > Subscriptions

- > Login/Group Roles
- > Tablespaces

- Use the [Query tool](#) button to open the Query tool in the current database context.
- Use the [View Data](#) button to view/edit the data stored in a selected table.
- Use the [Filtered Rows](#) button to access the Data Filter popup to apply a filter to a set of data for viewing/editing.

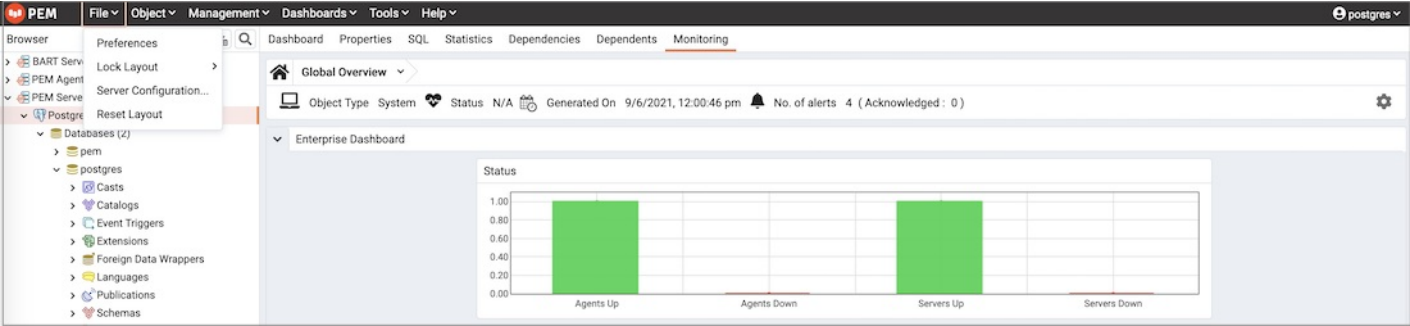
34.3.3 The PEM Menu Bar

The PEM menu bar provides access to commands and features that you can use to manage your database servers and the objects that reside on those servers. If an option is disabled:

- The database server to which you are currently connected may not support the selected feature.
- The selected menu option may not be valid for the current object (by design).
- The role that you have used to connect to the server may have insufficient privileges to change the selected object.

Context-sensitive menus across the top of the PEM web interface allow you to customize your environment and provide access to the enterprise management features of PEM.

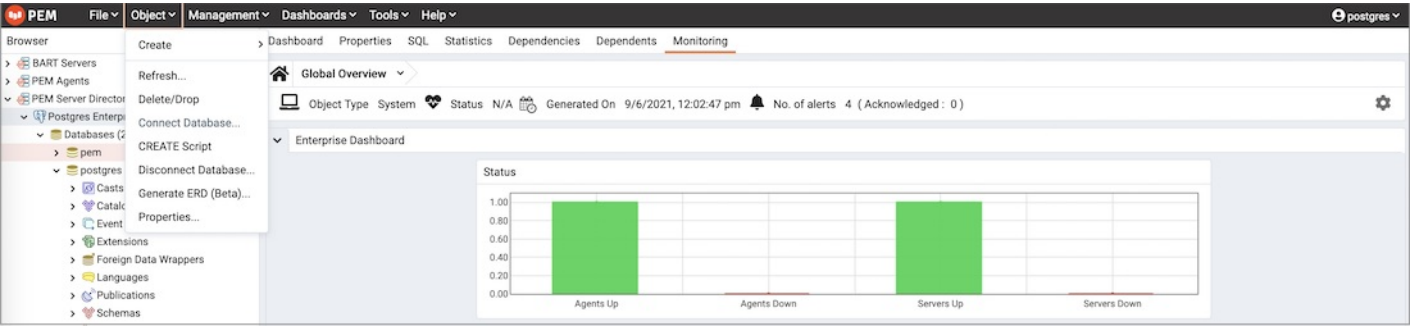
The File Menu



Use the [File](#) menu to access the following options:

| Menu Option          | Action  |
|----------------------|---|
| Preferences          | Click to open the <a href="#">Preferences</a> dialog to customize your PEM client settings.   |
| Lock Layout          | Click to open a sub-menu to select the level for locking the UI layout. This can also be changed from the Browser -> Display settings tab <a href="#">preferences</a> . |
| Server Configuration | Click to open the Server Configuration dialog and update your PEM server configuration settings.  |
| Reset Layout         | If a workspace panel is popped out by mistake or intentionally it can be reset back to default using Reset Layout.  |

The Object Menu



The [Object](#) menu is context-sensitive. Use the [Object](#) menu to access the following options:



| Menu Option                | Action  |
|----------------------------|---|
| Create                     | Click <b>Create</b> to access a context menu that provides context-sensitive selections. Your selection opens a <b>Create</b> dialog for creating a new object. |
| Refresh...                 | Click to refresh the currently selected object.   |
| Connect Server             | Click to open the <b>Connect to Server</b> dialog to establish a connection with a server.  |
| CREATE Script              | Click to open the <b>Query tool</b> to edit or view the selected script.  |
| Disconnect Server/Database | Click to disconnect the selected server.  |
| Remove Server              | Click to remove the selected server from the browser tree.  |
| BART                       | Click to access a context menu that provides options for removing BART configuration, taking a BART backup, or revalidate the BART configuration.               |
| Clear Saved Password       | If you have saved the database server password, click to clear the saved password. Enabled only after password is saved.  |
| Clear SSH Tunnel Password  | If you have saved the ssh tunnel password, click to clear the saved password. Enabled only after password is saved.   |
| Drop Cascade               | Click to delete the currently selected object and all dependent objects from the server.  |
| Properties...              | Click to review or modify the currently selected object's properties  |
| Delete/Drop                | Click to delete the currently selected object from the server.  |
| Connect Database           | Click to connect to selected database.  |
| Trigger(s)                 | Click to <b>Disable</b> or <b>Enable</b> trigger(s) for the currently selected table.   |
| Truncate                   | Click to remove all rows from a table (Truncate) or to remove all rows from a table and its child tables (Truncate Cascade).                                    |
| View/Edit Data             | Click to access a context menu that provides several options (All Rows, First 100 Rows, Last 100 Rows, Filtered Rows) for viewing data.                         |
| Count Rows                 | Click to count the number of rows of the selected table.  |
| Reset Statistics           | Click to reset the statistics of the selected table.  |
| Scripts                    | Click to CREATE, DELETE, INSERT, SELECT and UPDATE script for the selected table.   |

## The Management Menu

The screenshot displays the PEM interface. The 'Management' menu is open, showing various management tools. The background dashboard includes a 'Status' section with a bar chart showing 'Agents Up' and 'Servers Up' at 1.00, and a table of system metrics.

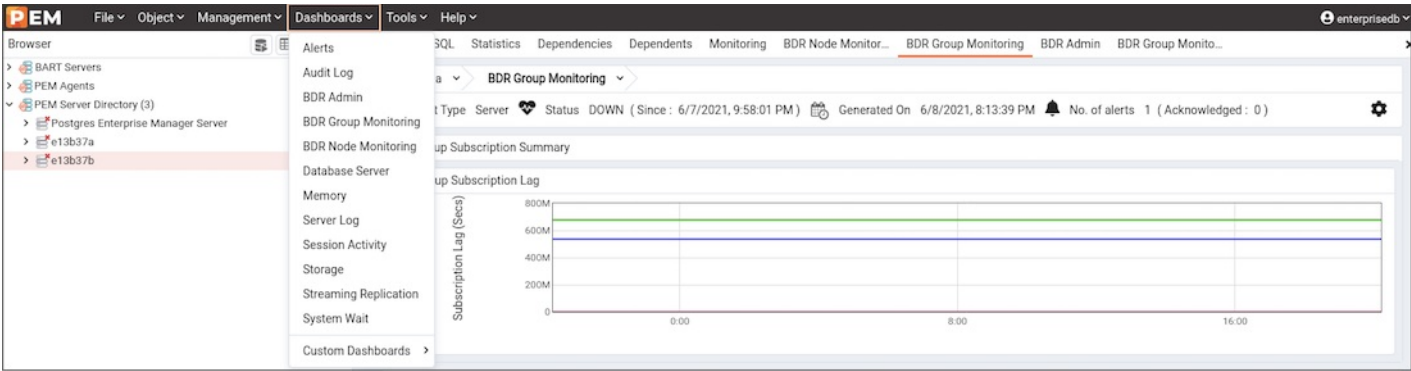
| Blackout                 | Status | Name                             | Alerts | Version | Processes | Threads | CPU Utilisation (%) | Memory Utilisation (%) | Swap Utilisation (%) | Disk Utilisation |
|--------------------------|--------|----------------------------------|--------|---------|-----------|---------|---------------------|------------------------|----------------------|------------------|
| <input type="checkbox"/> | UP     | Postgres Enterprise Manager Host | 1      | 8.0.1   | 388       | 953     | 1.48                | 45.93                  | 1.61                 | 24.91            |

Use the **Management** menu to access the following PEM features:

| Menu Option         | Action  |
|---------------------|---|
| Audit Manager...    | Click to open the <b>Audit Manager</b> and configure auditing on your monitored servers.                              |
| Auto Discovery...   | Click to open the <b>Auto Discovery</b> dialog to instruct a PEM agent to locate and bind monitored database servers. |
| Capacity Manager... | Click to open the <b>Capacity Manager</b> dialog and analyze historical or project future resource usage.             |
| Log Manager...      | Click to open the <b>Log Manager</b> dialog and configure log collection for a server.                                |
| Manage Alerts...    | Click to access the <b>Manage Alerts</b> tab and create or modify alerting behavior.                                  |

| Menu Option                     | Action  |
|---------------------------------|---|
| Manage Charts...                | Click to open the <a href="#">Manage Charts</a> tab to create or modify PEM charts.   |
| Manage Dashboards...            | Click to open the <a href="#">Manage Dashboards</a> dialog to VACUUM, ANALYZE, REINDEX, or CLUSTER.                                 |
| Manage Probes...                | Click to open the <a href="#">Manage Probes</a> dialog to VACUUM, ANALYZE, REINDEX, or CLUSTER.                                     |
| Postgres Expert...              | Click to open the <a href="#">Postgres Expert</a> wizard and perform a static analysis of your servers and databases.               |
| Postgres Log Analysis Expert... | Click to access the <a href="#">Postgres Log Analysis Expert</a> dialog analyze log file contents for usage trends.                 |
| Scheduled Tasks...              | Click to open the <a href="#">Scheduled Tasks</a> tab and review tasks that are pending or recently completed.                      |
| Schedule Alert Blackout...      | Click to open the <a href="#">Schedule Alert Blackout</a> dialog and schedule the alerts blackout for your servers and agents.      |
| Tuning Wizard...                | Click to open the <a href="#">Tuning Wizard</a> dialog to generate a set of tuning recommendations for your server.                 |
| Reports                         | Click to open the <a href="#">Reports</a> dialog to generate the system configuration report and core usage report for your server. |

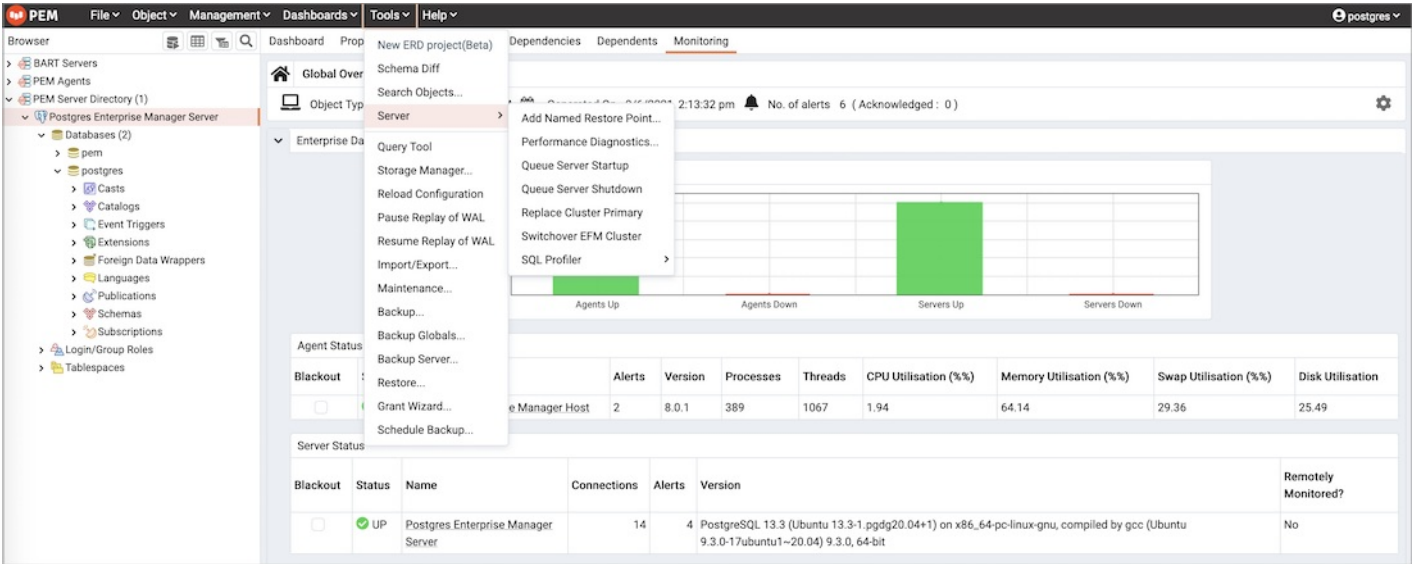
The Dashboards Menu



The [Dashboards](#) menu is context-sensitive; use the [Dashboards](#) menu to access the following options:

| Menu Option           | Action  |
|-----------------------|---|
| Alerts                | Click to open the <a href="#">Alerts Dashboard</a> for the selected node.                         |
| Audit Log             | Click to open the <a href="#">Audit Log Analysis Dashboard</a> for the selected node.             |
| PGD Admin             | Click to open the <a href="#">PGD Admin Dashboard</a> for the selected node.                      |
| PGD Group Monitoring  | Click to open the <a href="#">PGD Group Monitoring Dashboard</a> for the selected node.           |
| PGD Node Monitoring   | Click to open the <a href="#">PGD Node Monitoring Dashboard</a> for the selected node.            |
| Database Server       | Click to open the <a href="#">Database Analysis Dashboard</a> for the selected node.              |
| Memory                | Click to open the <a href="#">Memory Analysis Dashboard</a> for the selected node                 |
| Server Log            | Click to open the <a href="#">Server Log Analysis Dashboard</a> for the selected node.            |
| Session Activity      | Click to open the <a href="#">Session Activity Analysis Dashboard</a> for the selected node.      |
| Storage               | Click to open the <a href="#">Storage Analysis Dashboard</a> for the selected node.               |
| Streaming Replication | Click to open the <a href="#">Streaming Replication Analysis Dashboard</a> for the selected node. |
| System Wait           | Click to open the <a href="#">System Wait Analysis Dashboard</a> for the selected node.           |
| I/O Analysis          | Click to open the <a href="#">I/O Analysis Dashboard</a> for the selected node.                   |
| Object Activity       | Click to open the <a href="#">Object Activity Analysis Dashboard</a> for the selected node.       |
| Session Waits         | Click to open the <a href="#">Session Waits Analysis Dashboard</a> for the selected node.         |
| Operating System      | Click to open the <a href="#">Operating System Analysis Dashboard</a> for the selected node.      |
| Probe Log             | Click to open the <a href="#">Probe Log Analysis Dashboard</a> for the selected node.             |
| Custom Dashboards     | Click to open the Custom Dashboards. It will list custom dashboards configured by the user.       |

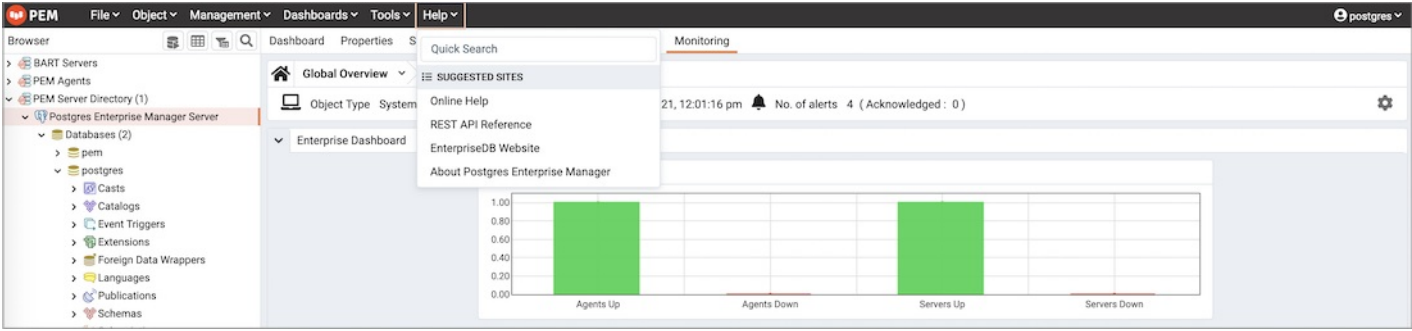
The Tools Menu



Use the **Tools** menu to access the following options:

| Menu Option           | Action  |
|-----------------------|---|
| New ERD Project(Beta) | Click to open the <b>ERD Tool</b> and start designing your database.  |
| Schema Diff           | Click to open the <b>Schema Diff</b> dialog to compare the schema objects between two database schemas.   |
| Search objects        | Click to open the Search Objects dialog to search the database objects within a database.   |
| Server                | Click to access the various server related tools such as Add Named Restore Point, Performance Diagnostics, Queue Server Startup, Queue Server Shutdown, Replace Cluster Primary, Switchover EFM Cluster and SQL Profiler. |
| Query tool            | Click to open the <b>Query tool</b> for the currently selected object.  |
| Storage Manager       | Click to open the <b>Storage manager</b> to upload, delete or download the backup files.  |
| Reload Configuration  | Click to update configuration files without restarting the server.  |
| Pause replay of WAL   | Click to pause the replay of the WAL log.   |
| Resume replay of WAL  | Click to resume the replay of the WAL log.  |
| Import/Export...      | Click to open the Import/Export data... dialog to import or export data from a table.   |
| Maintenance...        | Click to open the Maintenance... dialog to VACUUM, ANALYZE, REINDEX, or CLUSTER.  |
| Backup...             | Click to open the <b>Backup...</b> dialog to backup database objects.   |
| Backup Globals...     | Click to open the <b>Backup Globals...</b> dialog to backup cluster objects.  |
| Backup Server...      | Click to open the <b>Backup Server...</b> dialog to backup a server.  |
| Restore...            | Click to access the <b>Restore</b> dialog to restore database files from a backup.  |
| Grant Wizard...       | Click to access the <b>Grant Wizard</b> tool.   |
| Schedule Backup       | Click to access the Schedule Backup dialog for BART backups.  |

The Help Menu



Use the options on the **Help** menu to access online help documents or to review information about the PEM installation:

| Menu Option                       | Action  |
|-----------------------------------|---|
| Quick Search                      | Type your keywords in the Quick Search field. Typing atleast three characters will display all the matching possibilities under Menu items and the relevant documents under Help articles. Click on the options under Menu items to perform action of particular functionality or object. Click on any of the Help articles to open the help of that topic with highlighted text in a separate window. <b>Note:-</b> If any of the option under Menu items is disabled, then it will provide information via info icon. |
| Online Help                       | Click to open documentation for Postgres Enterprise Manager.  |
| REST API Reference                | Click to open the REST API Reference.   |
| EnterpriseDB Website              | Click to open the EnterpriseDB website in a browser window.   |
| About Postgres Enterprise Manager | Click to locate versioning and user information for Postgres Enterprise Manager.  |

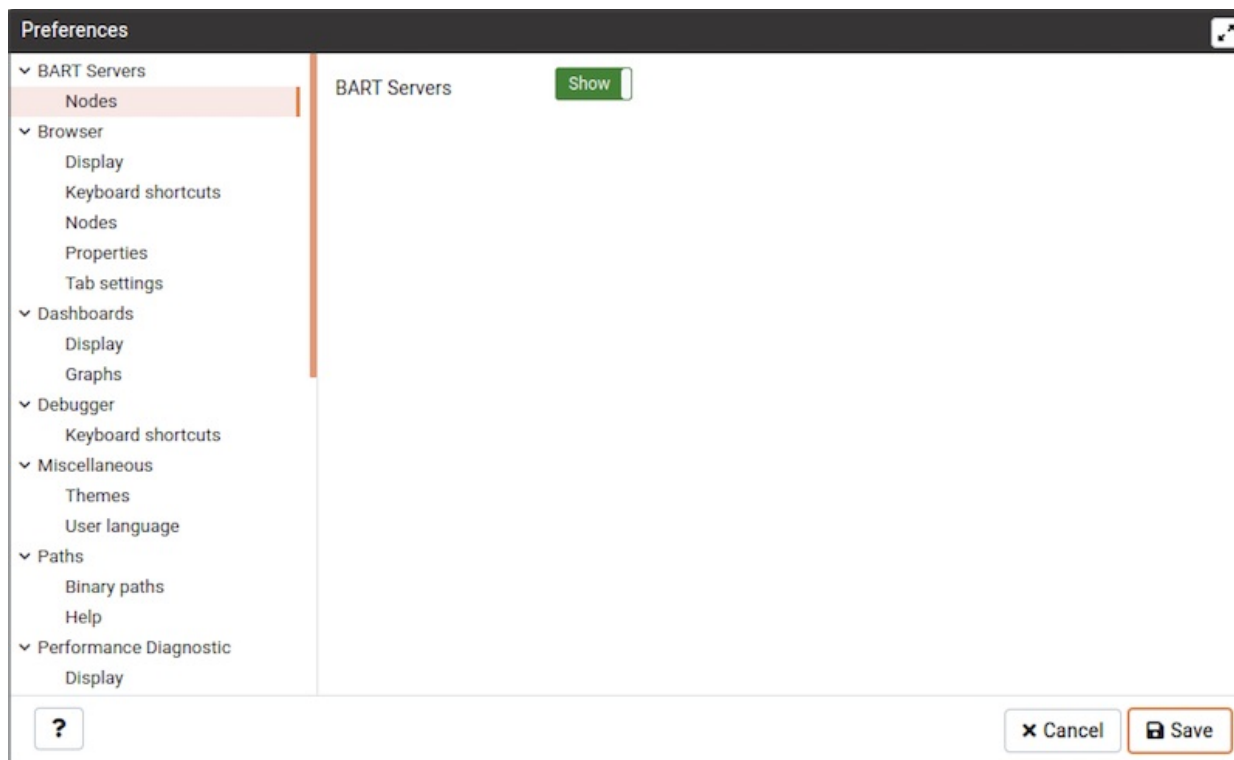
### 34.3.4 PEM Client Preferences

Use options on the **Preferences** dialog to customize the behavior of the PEM web interface. To open the dialog, select **Preferences** from the **File** menu. The left pane of the **Preferences** dialog displays a tree control; each node of the tree control provides access to options that are related to the node under which they are displayed.

- Use the plus sign (+) to the left of a node name to expand a segment of the tree control.
- Use the minus sign (-) to the left of a node name to close that node.

#### The BART Servers Node

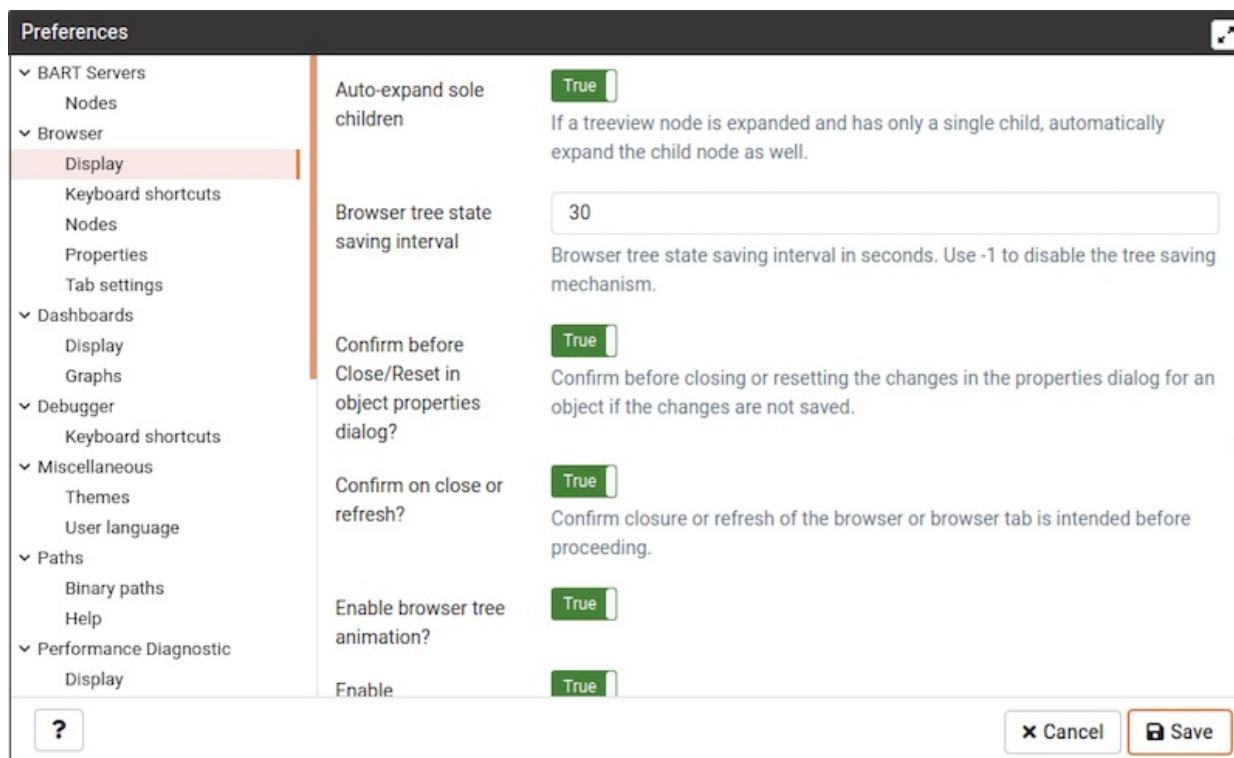
Use the **Nodes** panel to select the BART Servers that will be displayed in the Browser tree control of BART Servers:



- Slide the switch located next to BART Servers to show or hide the BART Servers in the browser tree.

### The Browser Node

Use the fields on the **Browser** node of the tree control to personalize your workspace.



Use the fields on the **Display** panel to specify general display preferences:

- When the **Enable browser tree animation?** switch is set to **True**, the client will display the animated tree control; if the switch is **False**, the tree control will be unanimated.
- When the **Auto-expand sole children** switch is set to **True**, child nodes will be automatically expanded if a treeview node is expanded

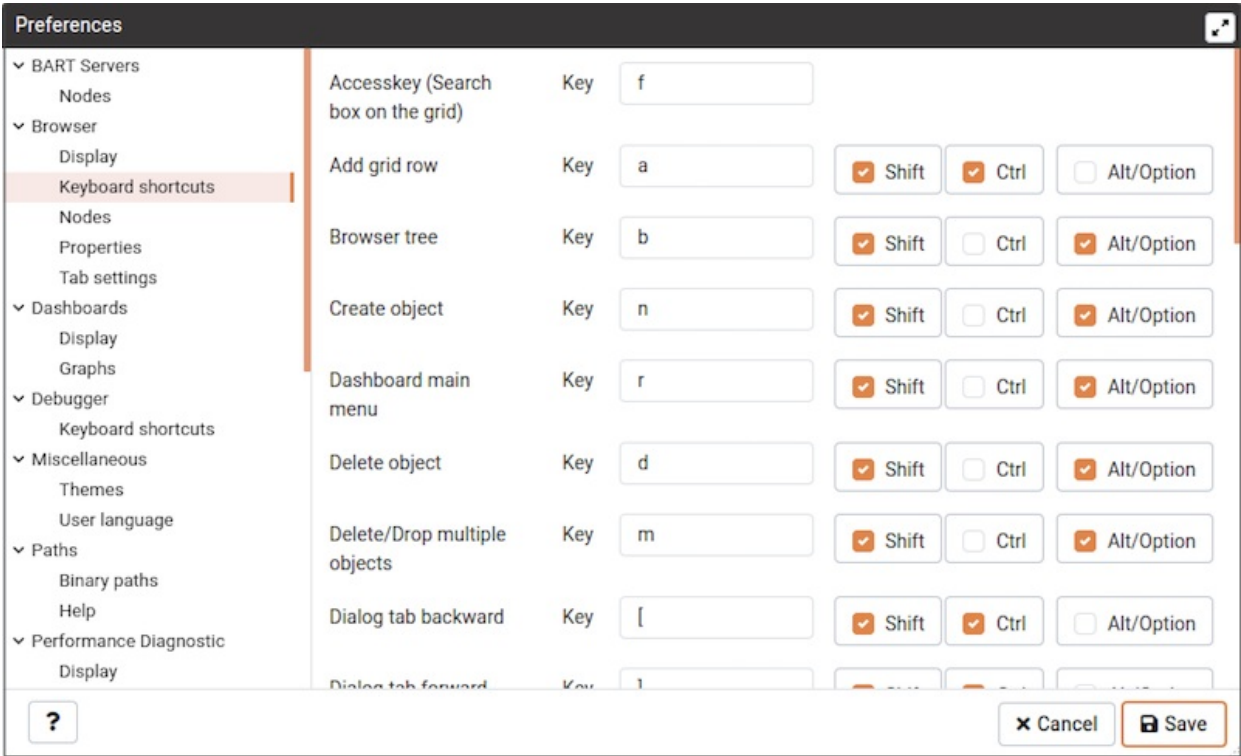
and has only a single child.

- Use the `Browser tree state saving interval` field to set the treeview state saving interval. A value of `-1` will disable the treeview state saving functionality.
- When the `Confirm before closing properties with unsaved changes` switch is set to `True`, PEM will warn you before closing the properties dialog of an object if there are any unsaved changes. On user confirmation, the properties dialog will close.
- When the `Confirm on close or refresh` switch is set to `True`, PEM will attempt to catch browser close or refresh events and prompt before allowing them to continue.
- When the `Show system objects?` switch is set to `True`, the client will display system objects such as system schemas (for example, `pg_temp`) or system columns (for example, `xmin` or `ctid`) in the tree control.
- When the `Enable dialogue/notification animation?` switch is set to `True`, the client will display the animated dialogues/notifications; if the switch is `False`, the tree control will be unanimated.
- Set `Show hidden groups?` to `True` to display hidden groups in the Browser tree control.
- Set `Show system objects?` to `True` to display system objects such as system schemas (for example, `pg_temp`) or system columns (for example, `xmin` or `ctid`) in the Browser tree control.
- Use the `Lock layout` field to lock the UI layout at different levels.

| Option          | Action  |
|-----------------|---|
| None            | No locking. Every panel is resizable and dockable.          |
| Prevent docking | This will disable the docking/undocking of the panels       |
| Full            | This will disable resizing, docking/undocking of the panels |

- When the `Show system objects?` switch is set to `True`, the client will display system objects such as system schemas (for example, `pg_temp`) or system columns (for example, `xmin` or `ctid`) in the tree control.

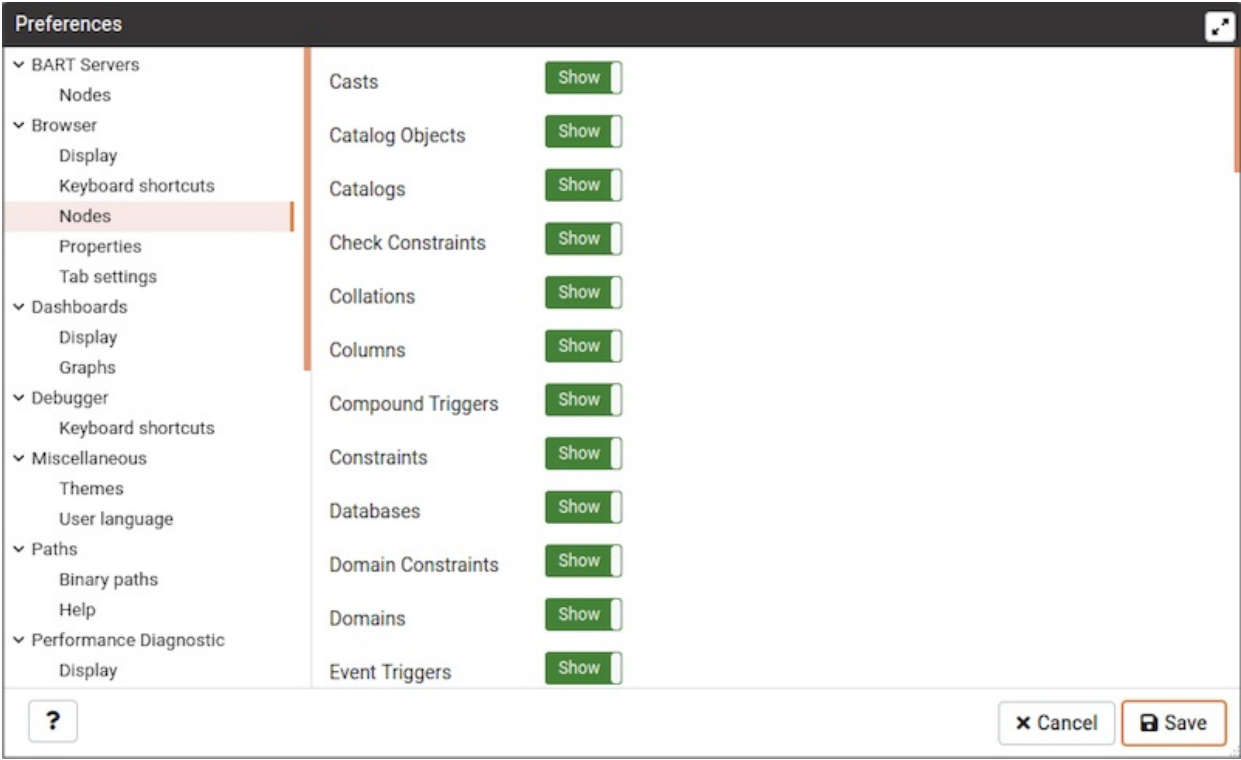
Use the fields on the `Keyboard shortcuts` panel to configure shortcuts for the main window navigation:



- Use controls on the `Keyboard shortcuts` panel to specify the combination of modifier keys that define shortcuts for the PEM main window.

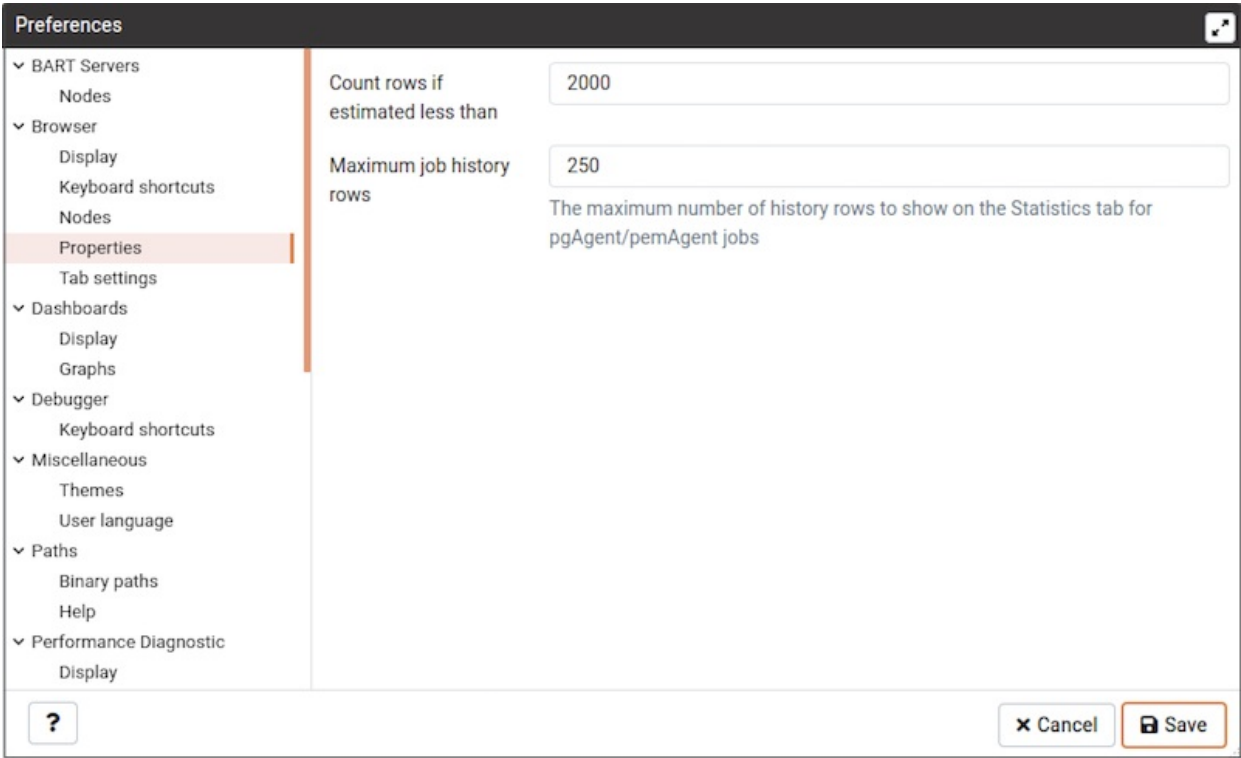
Use the fields on the `Nodes` panel to select the object types that will be displayed in the `Browser` tree control:





- The panel displays a list of database objects; slide the switch located next to each object type to **Show** or **Hide** the database object. When querying system catalogs, you can reduce the number of object types displayed to increase speed.

Use fields on the **Properties** panel to specify browser properties:



- Include a value in the **Count rows if estimated less than** field to perform a `SELECT count()` if the estimated number of rows in a table (as read from the table statistics) is below the specified limit. After performing the `SELECT count()`, PEM will display the row count. The default is 2000.
- Provide a value in the **Maximum job history rows** field to limit the number of rows to show on the statistics tab for pgAgent jobs. The default is 250.

Use field on **Tab settings** panel to specify the tab related properties.

**Preferences**

- ▼ BART Servers
  - Nodes
- ▼ Browser
  - Display
  - Keyboard shortcuts
  - Nodes
  - Properties
  - Tab settings**
- ▼ Dashboards
  - Display
  - Graphs
- ▼ Debugger
  - Keyboard shortcuts
- ▼ Miscellaneous
  - Themes
  - User language
- ▼ Paths
  - Binary paths
  - Help
- ▼ Performance Diagnostic
  - Display
- ▼ Query Tool
  - Auto completion
  - CSV/TXT Output
  - Display
  - Editor

**Debugger tab title**

Supported placeholders are %FUNCTION%, %ARGS%, %SCHEMA% and %DATABASE%. Users can provide any string with or without placeholders of their choice. The blank title will be revert back to the default title with placeholders.

**Dynamic tab size** ☐ False

If set to True, the tabs will take full size as per the title, it will also applicable for already opened tabs

**Open in new browser tab**

Select Query Tool, Debugger, or Schema Diff from the drop-down to set open in new browser tab for that particular module.

**Query tool tab title**

Supported placeholders are %DATABASE%, %USERNAME%, and %SERVER%. Users can provide any string with or without placeholders of their choice. The blank title will be revert back to the default title with placeholders.

**View/Edit data tab title**

Supported placeholders are %SCHEMA%, %TABLE%, %DATABASE%, %USERNAME%, and %SERVER%. Users can provide any string with or without placeholders of their choice. The blank title will be revert back to the default title with placeholders.

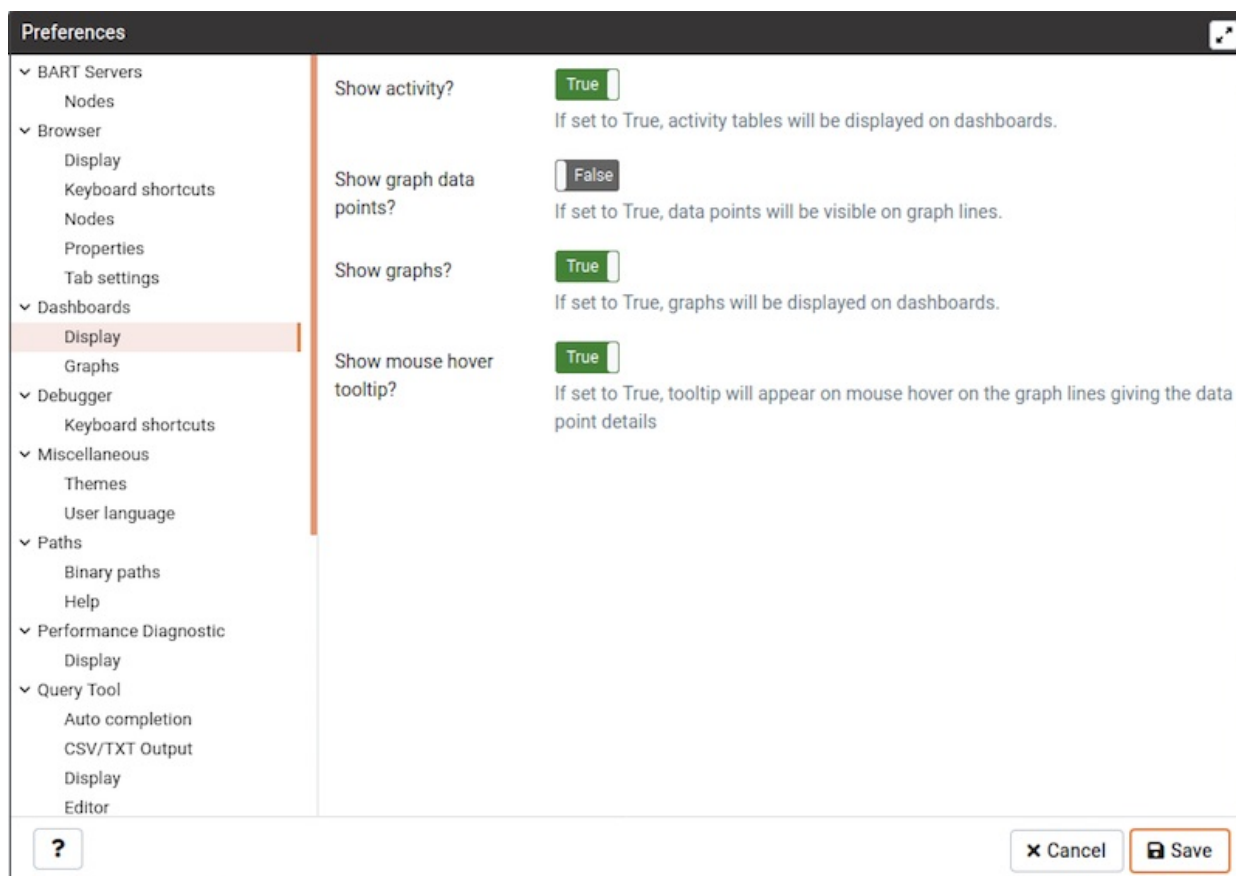
?

- Use *Debugger tab title placeholder* field to customize the Debugger tab title.
- When the *Dynamic tab size* If set to True, the tabs will take full size as per the title, it will also applicable for already opened tabs
- When the *Open in new browser tab* field is selected for Query tool, Schema Diff or Debugger, it will open in a new browser tab when invoked.
- Use the *Query tool tab title placeholder* field to customize the query tool tab title.
- Use *View/Edit tab title placeholder* field to customize the View/Edit Data tab title.

## The Dashboards Node

Expand the **Dashboards** node to specify your dashboard display preferences.





- Set the warning and alert threshold value to highlight the long-running queries on the dashboard.
- When the **Show activity?** switch is set to **True**, activity tables will be displayed on dashboards.
- When the **Show graph data points?** switch is set to **True**, data points will be visible on graph lines.
- When the **Show graphs?** switch is set to **True**, graphs will be displayed on dashboards.
- When the **Show mouse hover tooltip?** switch is set to **True**, a tooltip will appear on mouse hover on the graph lines giving the data point details.

Use the fields on the **Graphs** panel to specify your display preferences for the graphs on the **Dashboard** tab:

**Preferences**

- ▼ BART Servers
  - Nodes
- ▼ Browser
  - Display
  - Keyboard shortcuts
  - Nodes
  - Properties
  - Tab settings
- ▼ Dashboards
  - Display
  - Graphs**
- ▼ Debugger
  - Keyboard shortcuts
- ▼ Miscellaneous
  - Themes
  - User language
- ▼ Paths
  - Binary paths
  - Help
- ▼ Performance Diagnostic
  - Display
- ▼ Query Tool
  - Auto completion
  - CSV/TXT Output
  - Display
  - Editor

|                                     |   |  |
|-------------------------------------|---|--|
| Block I/O statistics refresh rate   | 1 | The number of seconds between graph samples. |
| Session statistics refresh rate     | 1 | The number of seconds between graph samples. |
| Transaction throughput refresh rate | 1 | The number of seconds between graph samples. |
| Tuples in refresh rate              | 1 | The number of seconds between graph samples. |
| Tuples out refresh rate             | 1 | The number of seconds between graph samples. |

? [Cancel] [Save]

Use the fields on the **Graphs** panel to specify your display preferences for the graphs on the **Dashboard** tab:

- Use the **Block I/O statistics refresh rate** field to specify the number of seconds between block I/O statistic samples displayed in graphs.
- Use the **Session statistics refresh rate** field to specify the number of seconds between session statistic samples displayed in graphs.
- Use the **Transaction throughput refresh rate** field to specify the number of seconds between transaction throughput samples displayed in graphs.
- Use the **Tuples in refresh rate** field to specify the number of seconds between tuples-in samples displayed in graphs.
- Use the **Tuples out refresh rate** field to specify the number of seconds between tuples-out samples displayed in graphs.

### The Debugger Node

Expand the **Debugger** node to specify your debugger display preferences.

Use the fields on the **Keyboard shortcuts** panel to configure shortcuts for the debugger window navigation:

Preferences

▼ BART Servers

Nodes

▼ Browser

Display

Keyboard shortcuts

Nodes

Properties

Tab settings

▼ Dashboards

Display

Graphs

▼ Debugger

Keyboard shortcuts

▼ Miscellaneous

Themes

User language

▼ Paths

Binary paths

Help

▼ Performance Diagnostic

Display

▼ Query Tool

Auto completion

CSV/TXT Output

Display

Editor

Accesskey (Clear all breakpoints)

Key

x

Accesskey (Continue/Start)

Key

c

Accesskey (Step into)

Key

i

Accesskey (Step over)

Key

o

Accesskey (Stop)

Key

s

Accesskey (Toggle breakpoint)

Key

t

Edit grid values

Key

q

☒ Shift

☐ Ctrl

☒ Alt/Option

Next tab

Key

]

☒ Shift

☐ Ctrl

☒ Alt/Option

Previous tab

Key

[

☒ Shift

☐ Ctrl

☒ Alt/Option

Switch Panel

Key

Tab

☒ Shift

☐ Ctrl

☒ Alt/Option

?

✕ Cancel

Save

The Miscellaneous Node

Expand the **Miscellaneous** node to specify miscellaneous display preferences.

Preferences

▼ BART Servers

Nodes

▼ Browser

Display

Keyboard shortcuts

Nodes

Properties

Tab settings

▼ Dashboards

Display

Graphs

▼ Debugger

Keyboard shortcuts

▼ Miscellaneous

Themes

User language

▼ Paths

Binary paths

Help

▼ Performance Diagnostic

Display

▼ Query Tool

Auto completion

CSV/TXT Output

Display

Editor

User language

English

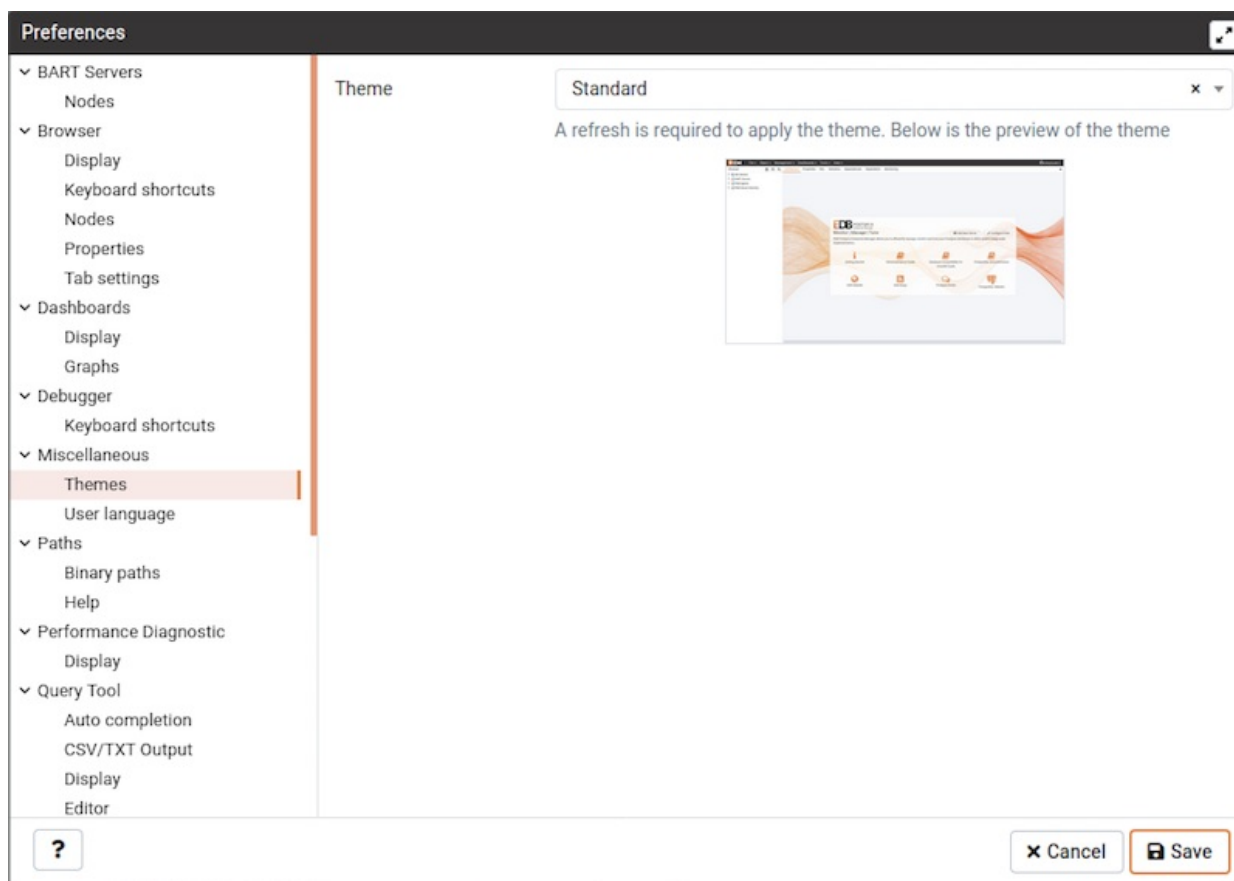
✕ ▼

?

✕ Cancel

Save

- Use the **User language** drop-down listbox to select the display language for the PEM web interface.



- Use the **Themes** drop-down listbox to select the theme for PEM. You'll also get a preview just below the drop down. Note that, to apply the theme you need to refresh the PEM page.

### The Paths Node

Expand the **Paths** node to specify the locations of supporting utility and help files.

**Preferences**

- ▼ BART Servers
  - Nodes
- ▼ Browser
  - Display
  - Keyboard shortcuts
  - Nodes
  - Properties
  - Tab settings
- ▼ Dashboards
  - Display
  - Graphs
- ▼ Debugger
  - Keyboard shortcuts
- ▼ Miscellaneous
  - Themes
  - User language
- ▼ **Paths**
  - Binary paths**
  - Help
- ▼ Performance Diagnostic
  - Display
- ▼ Query Tool
  - Auto completion
  - CSV/TXT Output
  - Display
  - Editor

**EDB Advanced Server Binary Path**  
  
 Path to the directory containing the EDB Advanced Server utility programs (pg\_dump, pg\_restore etc).

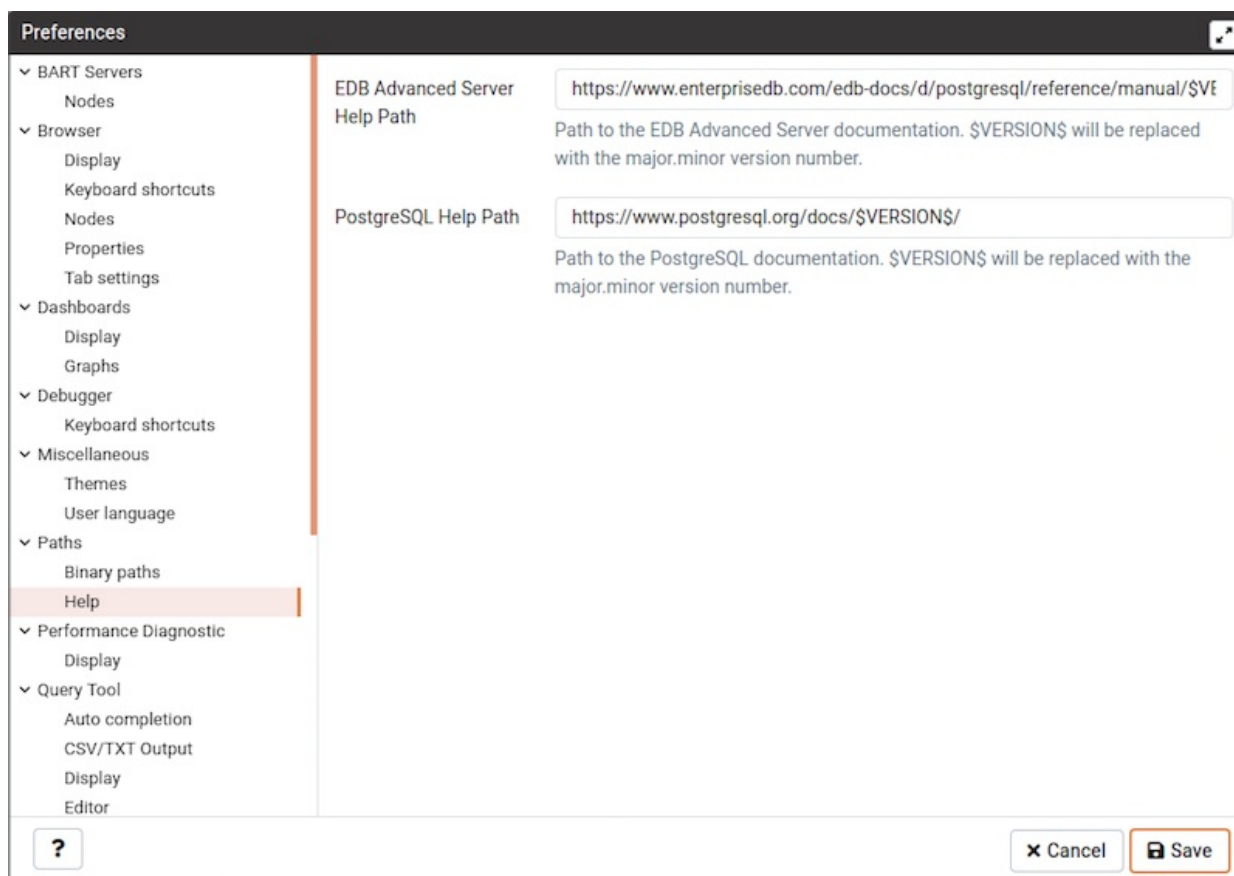
**Greenplum Database Binary Path**  
  
 Path to the directory containing the Greenplum Database utility programs (pg\_dump, pg\_restore etc).

**PostgreSQL Binary Path**  
  
 Path to the directory containing the PostgreSQL utility programs (pg\_dump, pg\_restore etc).

? Cancel Save

Use the fields on the **Binary paths** panel to specify the path to the directory that contains the utility programs (pg\_dump, pg\_restore, and pg\_dumpall) for monitored databases:

- Use the **EDB Advanced Server Binary Path** field to specify the location of the EDB Postgres Advanced Server utility programs. If this path is not set, PEM will attempt to find the utilities in standard locations used by EnterpriseDB.
- Use the **Greenplum Database Binary Path** field to specify the location of the Greenplum database utility programs. If this path is not set, PEM will attempt to find the utilities in standard locations used by Greenplum.
- Use the **PostgreSQL Binary Path** field to specify the location of the PostgreSQL utility programs. If this path is not set, PEM will attempt to find the utilities in standard locations used by PostgreSQL.



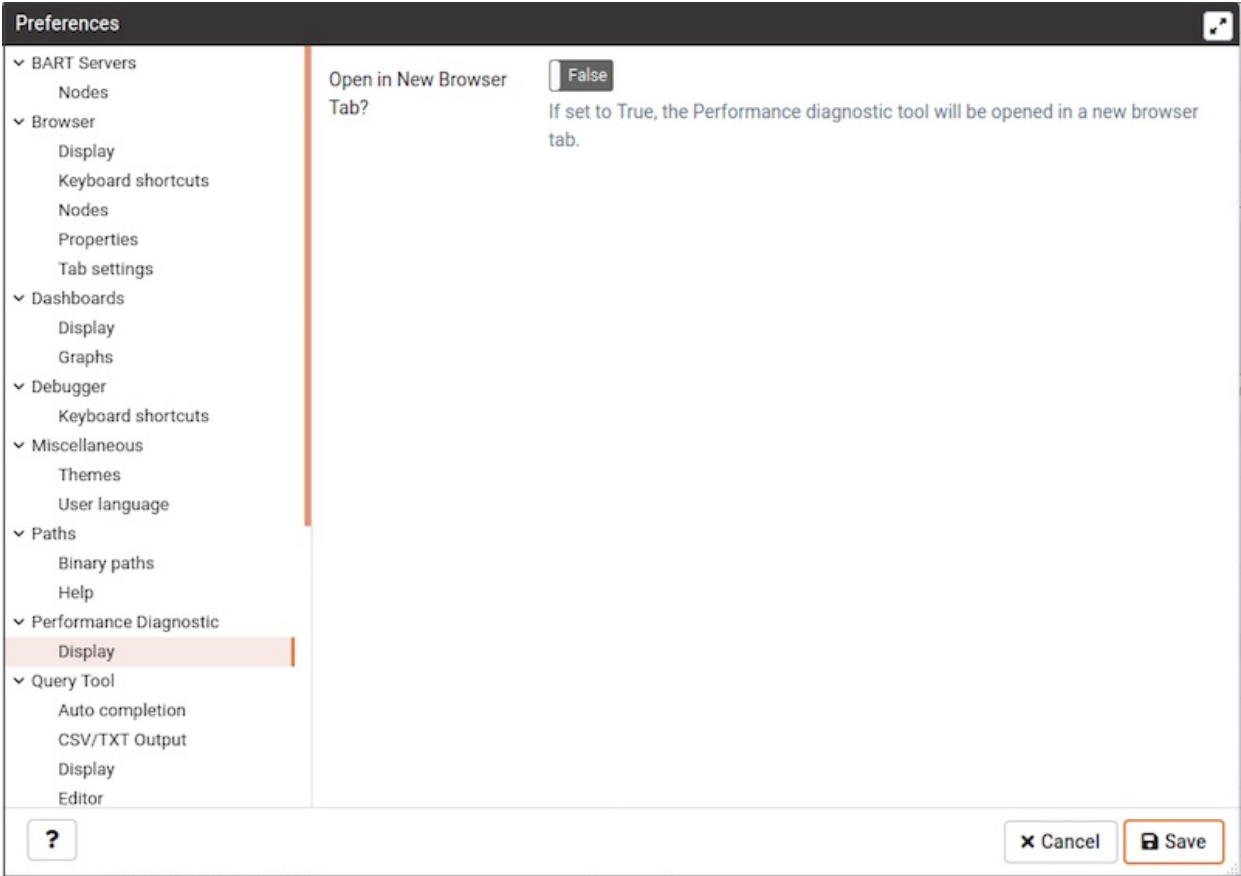
Use the fields on the **Help** panel to specify the location of help files.

- Use the **EDB Advanced Server Help Path** field to specify the path to EDB Postgres Advanced Server documentation.
- Use the **PostgreSQL Help Path** field to specify the path to PostgreSQL documentation.

Please note: the default help paths include the **VERSION** placeholder; the \$VERSION\$ placeholder will be replaced by the current database version.

### The Performance Diagnostic Node

Expand the **Performance Diagnostic** node to specify your preferences for the Performance Diagnostic tool.

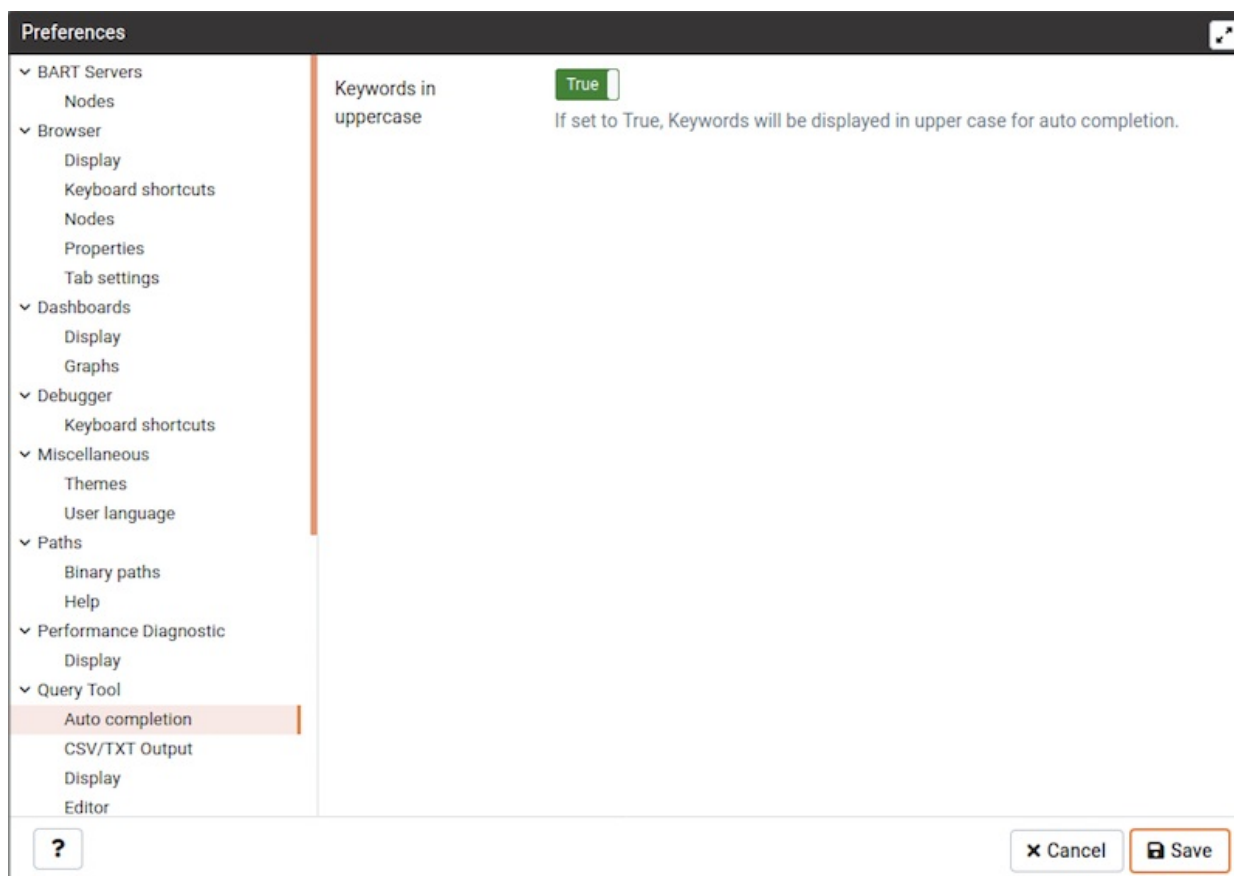


Use the fields on the **Performance Diagnostic** panel to control the Performance Diagnostic output.

- Use the **Default graph selection** field to specify the default selection range in hours for performance diagnostic graphs.
- When the **Open in new browser tab?** switch is set to True, the Performance Diagnostic tool will be opened in a new browser tab.

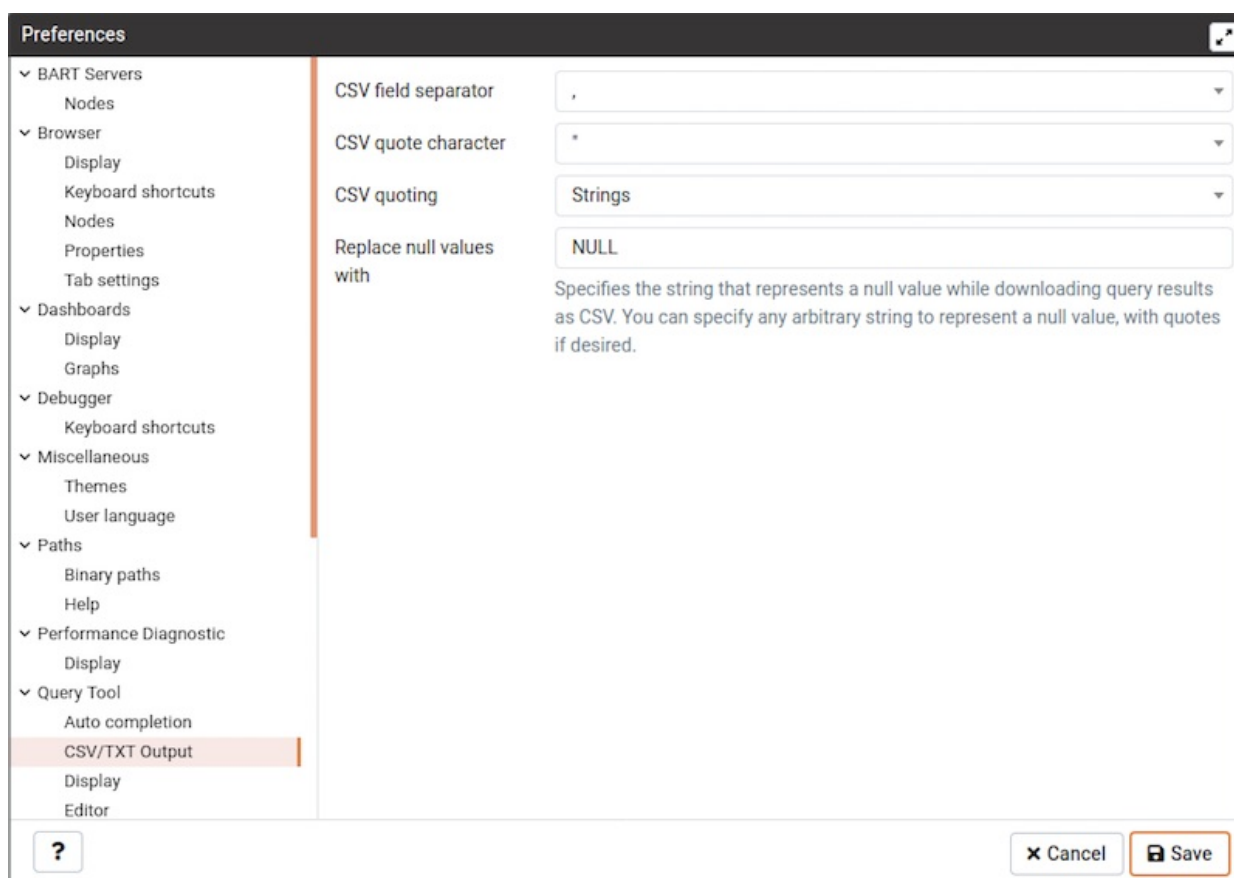
**The Query tool Node**

Expand the **Query tool** node to access panels that allow you to specify your preferences for the Query Editor tool.



Use the fields on the **Auto Completion** panel to set the auto completion options.

- When the **Keywords in uppercase** switch is set to **True**, keywords are displayed in upper case.





Use the fields on the **CSV Output** panel to control the CSV output.

- Use the **CSV field separator** drop-down listbox to specify the separator character that will be used in CSV/TXT output.
- Use the **CSV quote character** drop-down listbox to specify the quote character that will be used in CSV/TXT output.
- Use the **CSV quoting** drop-down listbox to select the fields that will be quoted in the CSV/TXT output; select **Strings** , **All**, or **None** .
- Use the **Replace null values with** option to replace null values with specified string in the output file. Default is set to 'NULL'.

Preferences

Themes

User language

▼ Paths

Binary paths

Help

▼ Performance Diagnostic

Display

▼ Query Tool

Auto completion

CSV/TXT Output

Display

Editor

Explain

Keyboard shortcuts

Options

Results grid

SQL formatting

▼ SQL Profiler

Display

Keyboard shortcuts

▼ Scheduled Tasks

Options

▼ Schema Diff

Display

▼ Storage

Options

Connection status

True

If set to True, the Query Tool will monitor and display the connection and transaction status.

Connection status refresh rate

2

The number of seconds between connection/transaction status polls.

Query info notifier timeout

5

The length of time to display the query info notifier after execution has completed. A value of -1 disables the notifier and a value of 0 displays it until clicked. Values greater than 0 display the notifier for the number of seconds specified.

?

Cancel

Save

Use the fields on the **Display** panel to specify your preferences for the Query tool display.

- When the **Connection status** switch is set to **True** , each new instance of the Query tool will display connection and transaction status.
- Use the **Connection status refresh rate** field to specify the number of seconds between connection/transaction status updates.
- Use the **Query info notifier timeout** field to control the behaviour of the notifier that is displayed when query execution completes. A value of **-1** will disable the notifier, and a value of 0 will display it until clicked. If a positive value above zero is specified, the notifier will be displayed for the specified number of seconds. The default is **5** .

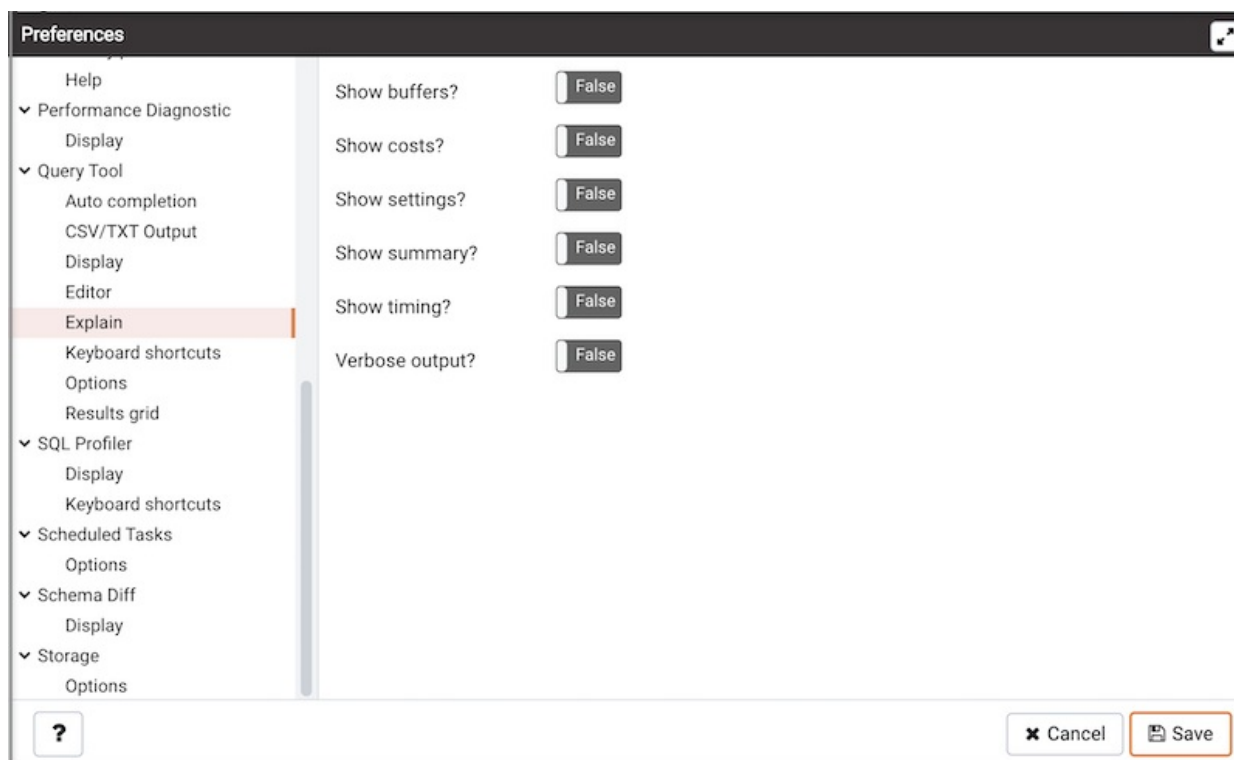
The screenshot shows the 'Preferences' dialog box with the 'Editor' panel selected in the left sidebar. The main area displays five settings for the query editor:

- Brace matching?**: A toggle switch set to 'True'. Description: 'Specifies whether or not to highlight matched braces in the editor.'
- Code folding?**: A toggle switch set to 'True'. Description: 'Enable or disable code folding. In plain text mode, this will have no effect as code folding is always disabled in that mode. Disabling will improve editor performance with large files.'
- Font size**: A text input field containing the value '1'. Description: 'The font size to use for the SQL text boxes and editors. The value specified is in "em" units, in which 1 is the default relative font size. For example, to increase the font size by 20 percent use a value of 1.2, or to reduce by 20 percent, use a value of 0.8. Minimum 0.1, maximum 10.'
- Insert bracket pairs?**: A toggle switch set to 'True'. Description: 'Specifies whether or not to insert paired brackets in the editor.'
- Line wrapping?**: A toggle switch set to 'False'. Description: 'Specifies whether or not to wrap SQL code in the editor.'
- Plain text mode?**: A toggle switch set to 'False'. Description: 'When set to True, keywords won't be highlighted and code folding will be disabled. Plain text mode will improve editor performance with large files.'

At the bottom of the dialog, there is a help icon (question mark), a 'Cancel' button, and a 'Save' button.

Use the fields on the **Editor** panel to change settings of the query editor.

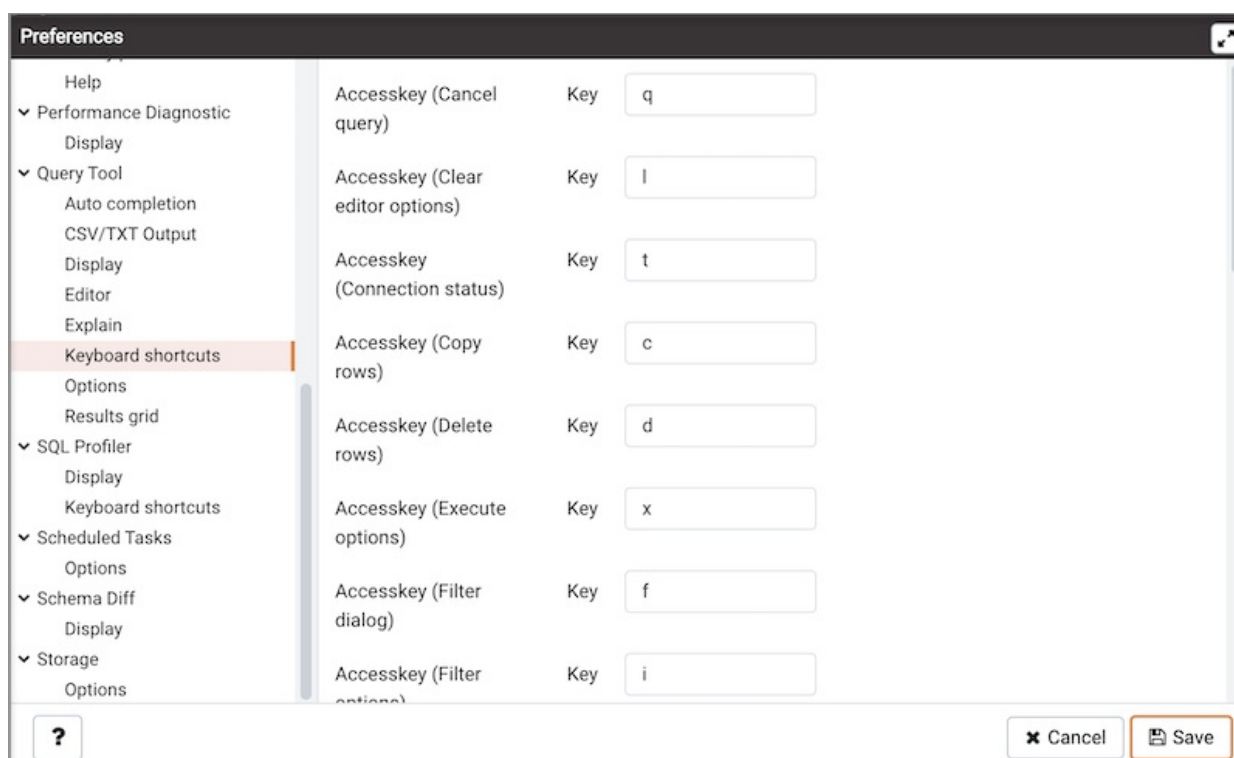
- When the **Brace matching?** switch is set to **True**, the editor will highlight pairs of matched braces.
- When the **Code folding?** switch is set to **False**, the editor will disable code folding. Disabling will improve editor performance with large files.
- Use the **Font size** field to specify the font size that will be used in text boxes and editors.
- When the **Insert bracket pairs?** switch is set to **True**, the editor will automatically insert paired brackets.
- When the **Line wrapping** switch is set to **True**, the editor will implement line-wrapping behavior.
- When the **Plain text mode?** switch is set to **True**, the editor mode will be changed to text/plain. Keyword highlighting and code folding will be disabled. This will improve editor performance with large files.



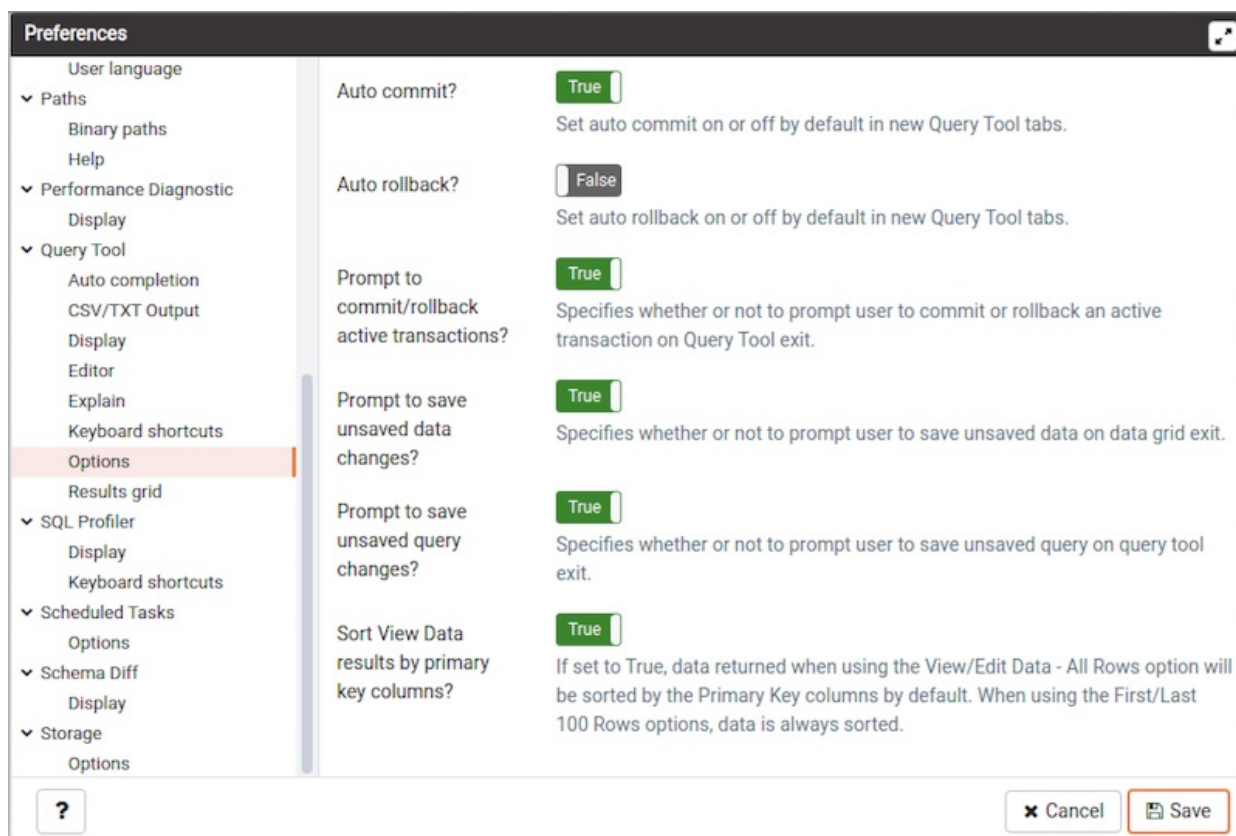
Use the fields on the **Explain** panel to specify the level of detail included in a graphical EXPLAIN.

- When the **Show Buffers?** switch is set to **True**, graphical explain details will include information about buffer usage.
- When the **Show Costs?** switch is set to **True**, graphical explain details will include information about the estimated startup and total cost of each plan, as well as the estimated number of rows and the estimated width of each row.
- When the **Show Timing?** switch is set to **True**, graphical explain details will include the startup time and time spent in each node in the output.
- When the **Verbose output?** switch is set to **True**, graphical explain details will include extended information about the query execution plan.

Use the fields on the **Keyboard shortcuts** panel to configure shortcuts for the Query tool.



Use the fields on the **Options** panel to manage Query tool preferences.



- When the **Auto-Commit?** switch is set to **True**, each successful query is committed after execution.
- When the **Auto-Rollback?** switch is set to **True**, failed queries are rolled back.
- When the **Copy SQL from main window to query tool?** switch is set to **True**, the CREATE sql of the selected object will be copied to query tool when query tool will open.
- When the **Prompt to save unsaved data changes?** switch is set to **True**, the editor will prompt the user to saved unsaved data when exiting the data editor.
- When the **Prompt to save unsaved query changes?** switch is set to **True**, the editor will prompt the user to saved unsaved query modifications when exiting the query tool.
- When the **Prompt to commit/rollback active transactions?** switch is set to **True**, the editor will prompt the user to commit or rollback changes when exiting the Query tool while the current transaction is not committed.
- When the **Sort View Data results by primary key columns?** If set to **True**, data returned when using the View/Edit Data - All Rows option will be sorted by the Primary Key columns by default. When using the First/Last 100 Rows options, data is always sorted.

**Preferences**

- Paths
  - Binary paths
  - Help
- Performance Diagnostic
  - Display
- Query Tool
  - Auto completion
  - CSV/TXT Output
  - Display
  - Editor
  - Explain
  - Keyboard shortcuts
  - Options
  - Results grid
  - SQL formatting**
- SQL Profiler
  - Display
  - Keyboard shortcuts
- Scheduled Tasks
  - Options
- Schema Diff
  - Display

Comma-first notation? ☐ False  
If set to True, comma-first notation for column names is used.

Identifier case ☒ Upper case ☐ Lower case ☐ Capitalized  
Convert identifiers to upper, lower, or capitalized casing.

Keyword case ☒ Upper case ☐ Lower case ☐ Capitalized  
Convert keywords to upper, lower, or capitalized casing.

Re-indent aligned? ☐ False  
If set to True, the indentations of the statements are changed, and statements are aligned by keywords.

Re-indent? ☒ True  
If set to True, the indentations of the statements are changed.

Spaces around operators? ☒ True  
If set to True, spaces are used around all operators.

?

Use the fields on the *SQL formatting* panel to specify your preferences for reformatting of SQL.

- Use the *Command-first notation* option to specify whether to place commas before or after column names.
- Use the *Identifier case* option to specify whether to change identifiers (object names) into upper, lower, or capitalized case.
- Use the *Keyword case* option to specify whether to change keywords into upper, lower, or capitalized case.
- Use the *Re-indent aligned?* option to specify that indentations of statements should be changed, aligned by keywords.
- Use the *Re-indent?* option to specify that indentations of statements should be changed.
- Use the *Spaces around operators?* option to specify whether or not to include spaces on either side of operators.
- Use the *Strip comments?* option to specify whether or not comments should be removed.
- Use the *Tab size* option to specify the number of spaces per tab or indent.
- Use the *Use spaces?* option to select whether to use spaces or tabs when indenting.
- Use the *Wrap after N characters* option to specify the column limit for wrapping column separated lists (e.g. of column names in a table). If set to 0 (zero), each item will be on it's own line.

**Preferences**

- Help
- Performance Diagnostic
  - Display
- Query Tool
  - Auto completion
  - CSV/TXT Output
  - Display
  - Editor
  - Explain
  - Keyboard shortcuts
  - Options
  - Results grid
- SQL Profiler
  - Display
  - Keyboard shortcuts
- Scheduled Tasks
  - Options
- Schema Diff
  - Display
- Storage
  - Options

Result copy field separator

Result copy quote character

Result copy quoting

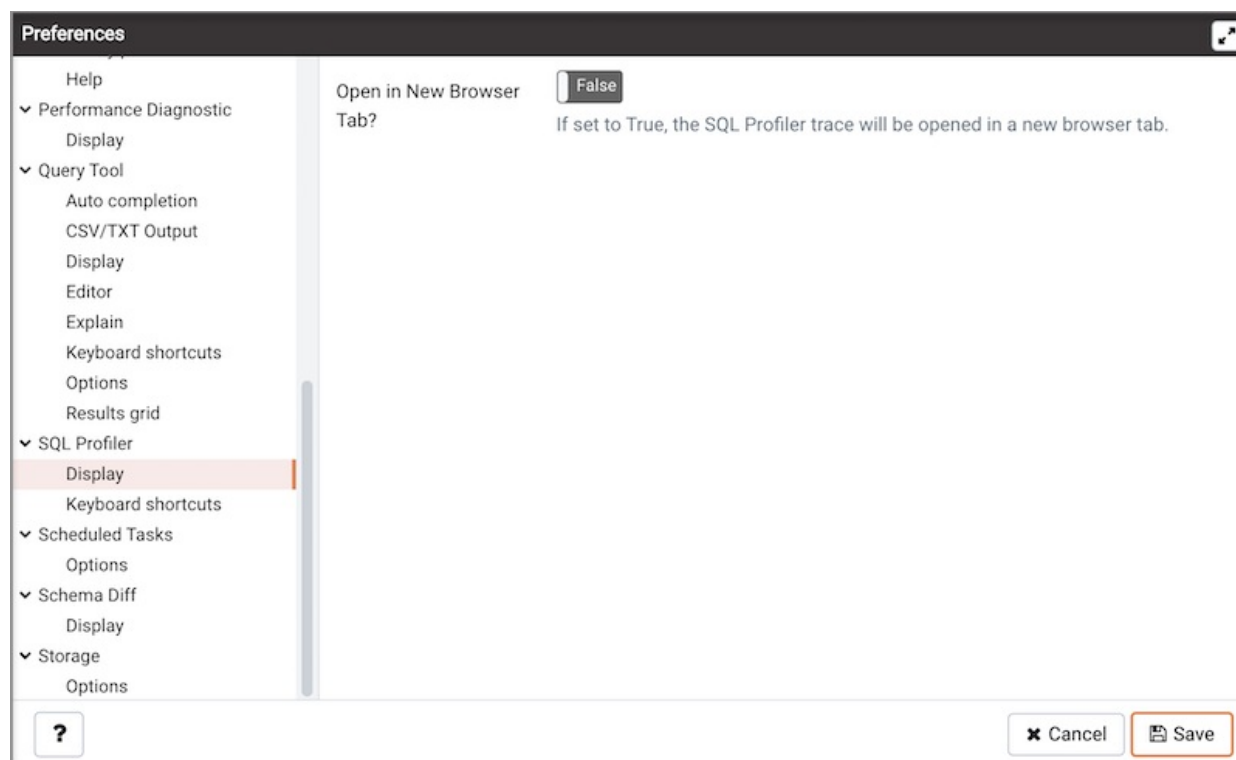
?

Use the fields on the **Results grid** panel to specify your formatting preferences for copied data.

- When the *Columns sized by* is set to *Column data*, then data columns will auto-size to the maximum width of the data in the column as loaded in the first batch. If set to *Column name*, the column will be sized to the widest of the data type or column name.
- Specify the maximum width of the column in pixels when 'Columns sized by' is set to *Column data*. If 'Columns sized by' is set to *Column name* then this setting won't have any effect.
- Use the **Result copy field separator** drop-down listbox to select the field separator for copied data.
- Use the **Result copy quote character** drop-down listbox to select the quote character for copied data.
- Use the **Result copy quoting** drop-down listbox to select which type of fields require quoting; select **All**, **None**, or **Strings**.

## The SQL Profiler Node

Use fields on the **Display** panel to specify SQL Profiler preferences.



Set **Open in New Browser Tab?** to **True** to open SQL Profiler in a new browser tab when SQL Profiler is invoked.

Use the fields on the Keyboard shortcuts panel to configure shortcuts for toolbar buttons on SQL profiler trace window.



Preferences

Help

Performance Diagnostic

Display

Query Tool

Auto completion

CSV/TXT Output

Display

Editor

Explain

Keyboard shortcuts

Options

Results grid

SQL Profiler

Display

Keyboard shortcuts

Scheduled Tasks

Options

Schema Diff

Display

Storage

Options

Accesskey (Clear Trace)

Accesskey (Filter dialog)

Accesskey (Index Advisor)

Accesskey (Open menu)

Accesskey (Properties)

Accesskey (Refresh Trace)

Accesskey (Stop Trace)

Column Picker

Key

c

Key

t

Key

i

Key

o

Key

p

Key

r

Key

q

Key

w

?

Cancel

Save

The Scheduled Tasks Node

Use fields on the Options panel to specify Scheduled Tasks preferences.

Preferences

Help

Performance Diagnostic

Display

Query Tool

Auto completion

CSV/TXT Output

Display

Editor

Explain

Keyboard shortcuts

Options

Results grid

SQL Profiler

Display

Keyboard shortcuts

Scheduled Tasks

Options

Schema Diff

Display

Storage

Options

Auto refresh interval

0

Auto refresh interval in seconds. Set 0 to disabled auto refresh.

Number of characters

250

Number of characters to read from the backend for the job step output column

?

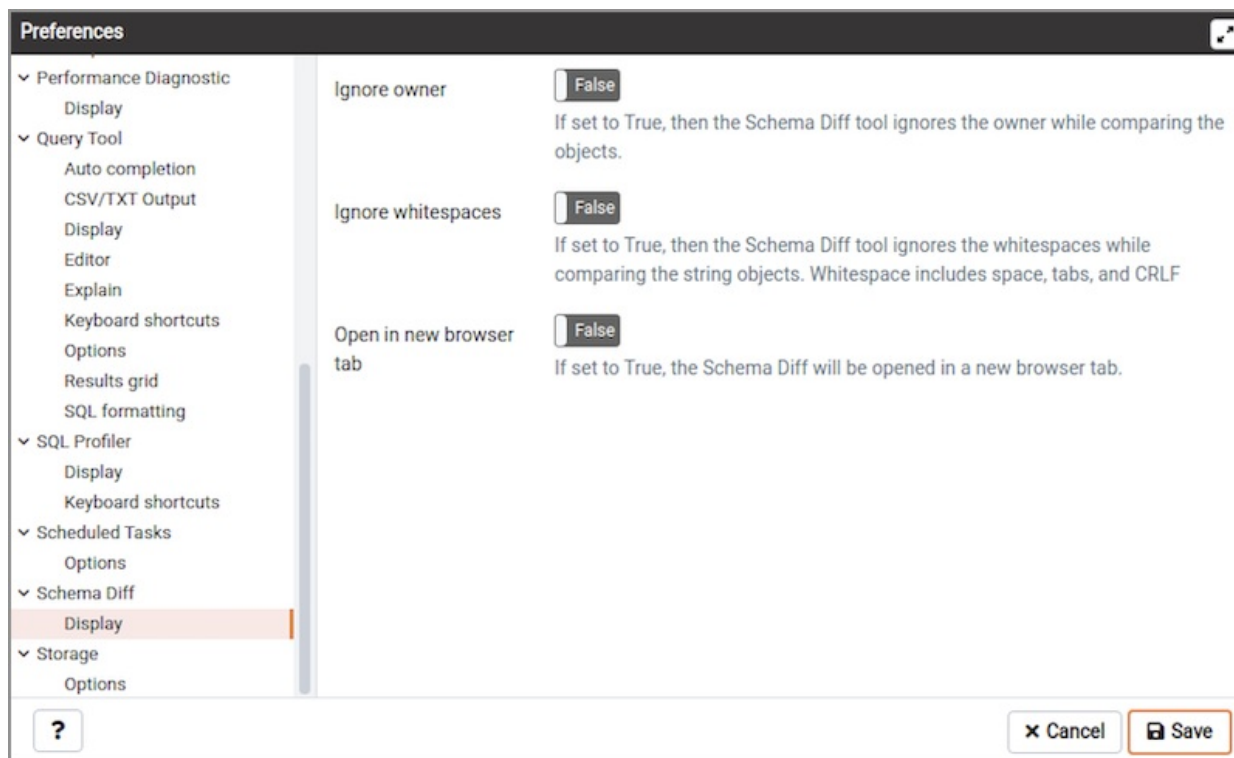
Cancel

Save

Use the Auto refresh interval field to specify the number of seconds between automatic refreshes; a value of 0 disables auto refresh.

The Schema Diff Node

Expand the Schema Diff node to specify your display preferences.

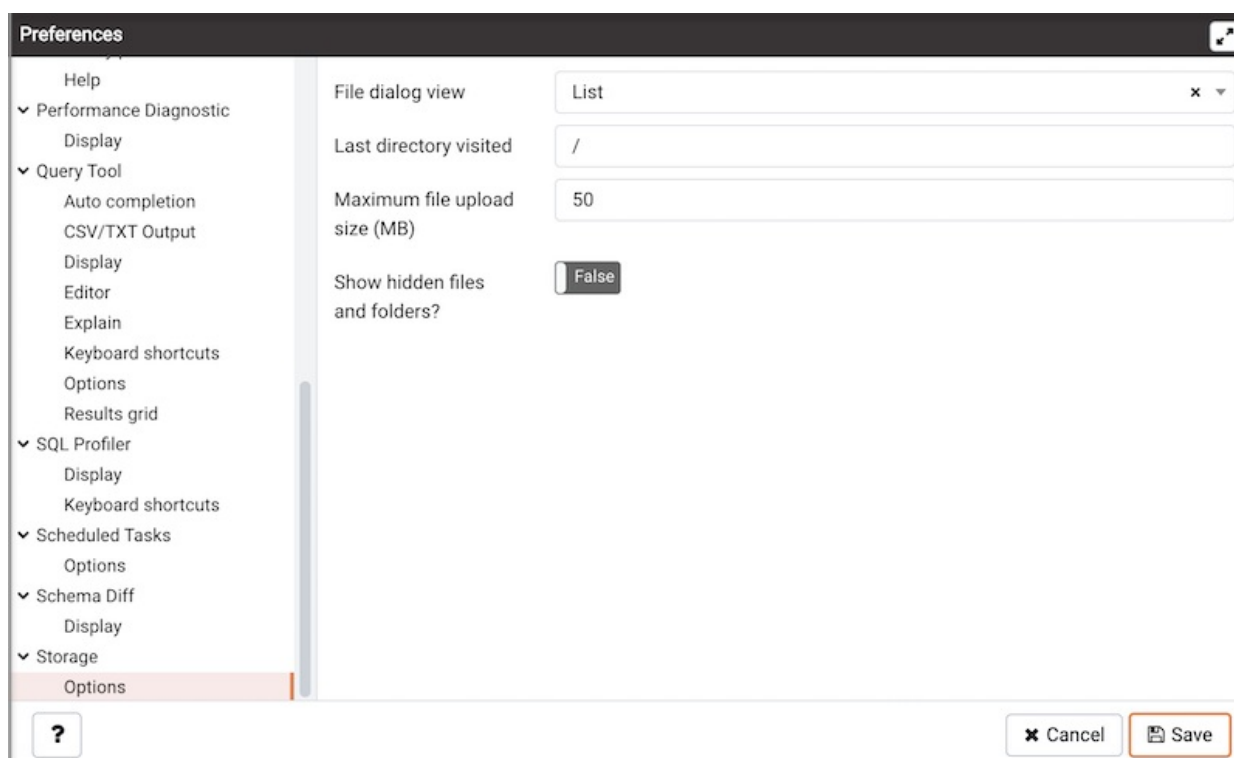


Use the *Ignore owner* switch to ignores the owner while comparing the objects.

Use the *Ignore whitespaces* switch to ignores the whitespaces while comparing the string objects. Whitespace includes space, tabs, and CRLF.

### The Storage Node

Expand the *Storage* node to specify your storage preferences.



Use the fields on the *Options* panel to specify storage preferences.



- Use the **File dialog view** drop-down listbox to select the style of icons and display format that will be displayed when you open the file manager; select **List** to display a list view, or **Grid** to display folder icons.
- Use the **Last directory visited** field to specify the name of the folder in which the file manager will open.
- Use the **Maximum file upload size(MB)** field on the **Options** panel of the **Storage** node to specify the maximum file size for an upload.
- When the **Show hidden files and folders?** switch is set to **True** , the file manager will display hidden files and folders.

34.3.5 Keyboard Shortcuts

Keyboard shortcuts are provided in PEM to allow easy access to specific functions. Alternate shortcuts can be configured through **File > Preferences** if desired.”

Main Browser Window

When using main browser window, the following keyboard shortcuts are available:

| Shortcut for all platforms | Function                                    |
|----------------------------|---|
| Alt+Shift+F                | Open the File menu                          |
| Alt+Shift+O                | Open the Object menu                        |
| Alt+Shift+L                | Open the Tools menu                         |
| Alt+Shift+H                | Open the Help menu                          |
| Alt+Shift+B                | Focus the browser tree                      |
| Alt+Shift+[                | Move tabbed panel backward                  |
| Alt+Shift+]                | Move tabbed panel forward                   |
| Alt+Shift+Q                | Open the Query tool in the current database |
| Alt+Shift+V                | View Data in the selected table/view        |
| Alt+Shift+C                | Open the context menu                       |
| Alt+Shift+N                | Create an object                            |
| Alt+Shift+E                | Edit object properties                      |
| Alt+Shift+D                | Delete the object                           |
| Alt+Shift+G                | Direct debugging                            |

Dialog Tabs

Use the shortcuts below to navigate the tabsets on dialogs:

| Shortcut for all platforms | Function            |
|----------------------------|---------------------|
| Control+Shift+[            | Dialog tab backward |
| Control+Shift+]            | Dialog tab forward  |

Property Grid Controls

Use the shortcuts below when working with property grid controls:

| Shortcut for all platforms | Function                                 |
|----------------------------|--|
| Control+Shift+A            | Add row in Grid                          |
| Tab                        | Move focus to the next control           |
| Shift+Tab                  | Move focus to the previous control       |
| Return                     | Pick the selected an item in a combo box |
| Control+Shift+A            | Add row in Grid                          |

## SQL Editors

When using the syntax-highlighting SQL editors, the following shortcuts are available:

| Shortcut (Windows/Linux) | Shortcut (Mac)       | Function                            |
|--------------------------|----------------------|-------------------------------------|
| Alt + Left               | Option + Left        | Move to the beginning of the line   |
| Alt + Right              | Option + Right       | Move to the end of the line         |
| Ctrl + Alt + Left        | Cmd + Option + Left  | Move left one word                  |
| Ctrl + Alt + Right       | Cmd + Option + Right | Move right one word                 |
| Ctrl + /                 | Cmd + /              | Comment selected code (Inline)      |
| Ctrl + .                 | Cmd + .              | Uncomment selected code (Inline)    |
| Ctrl + Shift + /         | Cmd + Shift + /      | Comment/Uncomment code (Block)      |
| Ctrl + a                 | Cmd + a              | Select all                          |
| Ctrl + c                 | Cmd + c              | Copy selected text to the clipboard |
| Ctrl + r                 | Cmd + r              | Redo last edit un-done              |
| Ctrl + v                 | Cmd + v              | Paste text from the clipboard       |
| Ctrl + z                 | Cmd + z              | Undo last edit                      |
| Tab                      | Tab                  | Indent selected text                |
| Shift + Tab              | Shift + Tab          | Un-indent selected text             |
| Alt + g                  | Option + g           | Jump (to line:column)               |
| Ctrl + Space             | Ctrl + Space         | Auto-complete                       |
| Ctrl + f                 | Cmd + f              | Find                                |
| Ctrl + g                 | Cmd + g              | Find next                           |
| Ctrl + Shift + g         | Cmd + Shift + g      | Find previous                       |
| Ctrl + Shift + f         | Cmd + Shift + f      | Replace                             |

## Query tool

When using the Query tool, the following shortcuts are available:

| Shortcut (Windows/Linux) | Shortcut (Mac) | Function      |
|--------------------------|----------------|---------------|
| F5                       | F5             | Execute query |

| Shortcut (Windows/Linux) | Shortcut (Mac)  | Function                  |
|--------------------------|-----------------|---------------------------|
| F6                       | F6              | Save data changes         |
| F7                       | F7              | EXPLAIN query             |
| Shift + F7               | Shift + F7      | EXPLAIN ANALYZE query     |
| F8                       | F8              | Execute query to CSV file |
| <accesskey> + o          | <accesskey> + o | Open file                 |
| <accesskey> + s          | <accesskey> + s | Save file                 |
| <accesskey> + n          | <accesskey> + n | Find option drop down     |
| <accesskey> + c          | <accesskey> + c | Copy row(s)               |
| <accesskey> + p          | <accesskey> + p | Paste row(s)              |
| <accesskey> + d          | <accesskey> + d | Delete row(s)             |
| <accesskey> + f          | <accesskey> + f | Filter dialog             |
| <accesskey> + i          | <accesskey> + i | Filter options drop down  |
| <accesskey> + r          | <accesskey> + r | Row limit                 |
| <accesskey> + q          | <accesskey> + q | Cancel query              |
| <accesskey> + l          | <accesskey> + l | Clear option drop down    |
| <accesskey> + x          | <accesskey> + x | Execute option drop down  |
| <accesskey> + t          | <accesskey> + t | Display connection status |
| <accesskey> + y          | <accesskey> + y | Copy SQL on history panel |

## Debugger

When using the Debugger, the following shortcuts are available:

| Shortcut (Windows/Linux) | Shortcut (Mac)     | Function                     |
|--------------------------|--------------------|------------------------------|
| <accesskey> + i          | <accesskey> + i    | Step in                      |
| <accesskey> + o          | <accesskey> + o    | Step over                    |
| <accesskey> + c          | <accesskey> + c    | Continue/Restart             |
| <accesskey> + t          | <accesskey> + t    | Toggle breakpoint            |
| <accesskey> + x          | <accesskey> + x    | Clear all breakpoints        |
| <accesskey> + s          | <accesskey> + s    | Stop                         |
| Alt + Shift + q          | Option + Shift + q | Enter or Edit values in Grid |

## Inner Tab and Panel Navigation

When using the Query tool and Debugger, the following shortcuts are available for inner panel navigation:

| Shortcut (Windows/Linux) | Shortcut (Mac)    | Function                            |
|--------------------------|-------------------|-------------------------------------|
| Alt + Shift + ]          | Alt + Shift + ]   | Move to next tab within a panel     |
| Alt + Shift + [          | Alt + Shift + [   | Move to previous tab within a panel |
| Alt + Shift + Tab        | Alt + Shift + Tab | Move between inner panels           |

## Access Key

<accesskey> is browser and platform dependant. The following table lists the default access keys for supported browsers.

| Browser           | Windows     | Linux       | Mac           |
|-------------------|-------------|-------------|---------------|
| Internet Explorer | Alt         | Alt         |               |
| Chrome            | Alt         | Alt         | Ctrl + Option |
| Firefox           | Alt + Shift | Alt + Shift | Ctrl + Option |
| Safari            | Alt         |             | Ctrl + Option |

### 34.3.6 Search objects

Search Objects - postgres/enterprisedb@Postgres Enterprise Manager Server

All types

| Object name                      | Type       | Browser path   |
|----------------------------------|------------|--|
| sslutils                         | Extensions | Extensions/sslutils                                  |
| openssl_csr_to_cert (...)        | Functions  | Schemas/public/Functions/openssl_csr_to_cert         |
| openssl_get_cert_expiry_dat...   | Functions  | Schemas/public/Functions/openssl_get_cert_expiry_... |
| openssl_is_cert_expire_on (...)  | Functions  | Schemas/public/Functions/openssl_is_cert_expire_on   |
| openssl_revoke_certificate (...) | Functions  | Schemas/public/Functions/openssl_revoke_certific...  |
| openssl_rsa_generate_crl (...)   | Functions  | Schemas/public/Functions/openssl_rsa_generate_crl    |
| openssl_rsa_generate_key (...)   | Functions  | Schemas/public/Functions/openssl_rsa_generate_k...   |
| openssl_rsa_key_to_csr (...)     | Functions  | Schemas/public/Functions/openssl_rsa_key_to_csr      |
| sslutils_version                 | Functions  | Schemas/public/Functions/sslutils_version            |

9 matches found.

With this dialog, you can search for almost any kind of objects in a database.

You can access it by right clicking a database or any of its child nodes and select "Search objects". You can also access it by hitting the shortcut (default ALT+SHIFT+S).

The minimum pattern length are 3 characters. The search performed is non-casesensitive and will find all objets whose name contains the pattern. You can only search for object names currently. Examples are: abc, %ab%, ab%c, %%%, etc.

The result is presented in the grid with object name, object type and the object tree path in the browser tree. You can double click on a result row to select the object in the browser tree. If the object is greyed out, this means that you have not enabled those object types in the [preferences](#), so you can't double click on it. You can click on the ellipsis appended to the function and procedure names to see there arguments.

You can filter based on a particular object type by selecting one from the object type dropdown. If the search button is hit when one of the object type is selected then only those types will be fetch from the database. An object type will not be visible in the dropdown if the database server does not support it or if it is not enabled from the [preferences](#).

## 34.4 Enterprise Management Features

Postgres Enterprise Manager offers a number of additional enterprise management features that will assist you in managing, analyzing, streamlining, and deploying Postgres functionality. PEM probes monitor managed servers, retrieving information that PEM then analyzes to create dashboards that display useful information and statistics about your hosts, servers and databases. PEM dialogs provide easy access to probe, server, and agent configurations so you can enable and customise the behaviour of PEM features.

- [Dashboards](#)
- [Server Configuration](#)

PEM configuration wizards simplify the process of configuring logging; after logs are gathered, you can use PEM features like the PEM Log Analysis Expert and Tuning Wizard to help you skillfully analyze log file contents. Postgres Expert analyses server configuration, and recommends changes that may improve performance, security, or reliability. Capacity Manager uses statistical data to analyze historical usage, and extrapolate future usage statistics for monitored objects.

- [Log Manager](#)
- [Audit Manager](#)
- [Postgres Log Analysis Expert](#)
- [Tuning Wizard](#)
- [Postgres Expert](#)
- [Capacity Manager](#)

PEM is highly customizable. You can modify system-defined alerts, or create custom alerts that notify a responsible administrator or group during a specific time period. PEM charts and dashboards are designed to provide an at-a-glance overview of the state of your systems, but you can design custom charts that display the most useful combinations of metric data returned by PEM probes. Then, create custom dashboards with details that [you](#) need to identify problems, plan for resource usage, or diagnose unique performance issues.

- [Alerting](#)
- [Using the Manage Charts Tab](#)
- [The PEM Manage Dashboards Tab](#)
- [The Manage Probes Tab](#)

PEM can schedule jobs on monitored servers. you can use the wizards to schedule the setup to occur at a time when server usage is low, or when a system restart won't interrupt your users. The Scheduled Task tab provides an overview of all the tasks that are pending execution or that have recently completed on your servers. Performance diagnostic feature helps you analyze the database performance for Advanced Server as well as PostgreSQL (from 8.0 onwards) instances by monitoring the wait events.

- [Schedule Alert Blackout](#)
- [Scheduled System Jobs](#)
- [Scheduled Task Tab](#)
- [Creating a PEM Scheduled Job](#)
- [Sending email notifications for a job](#)
- [Task Viewer](#)
- [Monitoring a Failover Manager Cluster](#)
- [Monitoring a PGD Group](#)
- [Performance Diagnostic](#)

PEM can generate system configuration and core usage reports of locally or remotely monitored servers in HTML as well as in JSON format.

- [Reports](#)

### 34.4.1 Dashboards

Postgres Enterprise Manager uses metrics (retrieved by probes) to generate the statistical information displayed on the dashboards. Dashboards are presented in a hierarchy comparable to the PEM client tree control; the dashboard for each object within the tree control displays the information for that object, as well as for any monitored object that resides below that level in the tree control, if appropriate.

Each dashboard header displays the date and time that the server was started (if relevant), the date and time that the dashboard was last updated, and the current number of triggered alerts. Navigation menus displayed in the dashboard header provide easy access to other dashboards. Menus are organised hierarchically; only those menus appropriate for the object currently highlighted in the tree control are available:

- Select **Global Overview** from any dashboard to return to the Global Overview dashboard.
- Select the name of an agent from the **Agents** menu to navigate to the Operating System Analysis dashboard for that agent.
- Select a server name from the **Servers** menu to navigate to the Server Analysis dashboard for that server.
- Select a database name from the **Databases** menu to navigate to the Database Analysis dashboard for that database.
- Use the **Dashboards** menu to navigate to informational dashboards at the global level, or for the selected agent, server or database.

Dashboards display statistical information in the form of:

- Tables - Tables provide statistical information collected by a PEM probe.
- Pie charts - Pie charts display information collected by the most recent execution of a probe.
- Bar graphs - Bar graphs display comparative statistics collected by the most recent execution of a probe.
- Line graphs - Line graphs display statistical data collected by PEM probes.

Options on the **Dashboard Configuration** dialog allow you to link the time lines of all of the line graphs on the dashboard. To open the **Dashboard Configuration** dialog, click the wrench icon displayed in the dashboard header.

- Set the **Link timelines of all the line charts** slider to **Enable** to indicate that the specified timeline should be applied to line graphs displayed on the dashboard; if set to **Disable**, your preferences will be preserved for later use, but will not modify the amount of data displayed.
- Use the **Days** selector to specify the number of days of gathered data that should be displayed on line graphs.
- Use the **Hour(s)** selector to specify the number of hours of gathered data that should be displayed on line graphs.
- Check the box next to **Remember configuration for this dashboard** to indicate that the customized time span should be applied to the current dashboard only; if left unchecked, the time span will be applied globally to line graphs on all dashboards.

Please note that settings specified on the **Dashboard Configuration** dialog are applied only to the current user's session.

When you've specified your preferences, click **Save** to preserve your changes and exit the dialog; click **Cancel** to exit the dialog without preserving your changes.

To sort statistics that are provided in table form, click on a column heading; click again to reverse the sort order. Each table offers a stable sort feature - For example, to sort a table by ascending **Session ID** within each user name group, sort first by the **Session ID** column, then sort by the **User Name** column.

Hover your mouse over the upper-right corner of each graph, chart or table to reveal the PEM client toolbar icons. Hover over an icon to display a tooltip that briefly explains the icon's functionality:

- Use the **Refresh** icon to update the information displayed on a dashboard.

- Use the **Save Chart as Image** icon to save the selected chart as a .jpeg image.
- Use the **Full Screen** icon to enlarge the chart to reveal granular details about the charted data.
- Click the **Personalize the chart configuration** icon to access a control panel that allows you to select chart-specific display details.
- Hover over the **Explain** icon to review a description of the information shared in the graph or chart.

In the lower-right corner of each graph or chart is a legend that identifies each item plotted in the graph or chart.

If displayed, click the information icon in the upper-left hand corner of a chart to display a note about the chart content, and if applicable, a link that will allow you to enable one or more probes that retrieve content for the chart.

## Accessing Dashboards

Navigation menus in the dashboard header provide easy access to other dashboards. The menus are organized hierarchically, allowing you to jump from object to object at any level:

- The **Global Overview** option opens the **Global Overview** dashboard.
- The **Agents** menu expands to display a list of agents. Select an agent from the list to access the **Operating System Analysis** dashboard for that agent.
- The **Servers** menu expands to display a list of monitored servers. Select a server from the list to access the **Server Analysis** dashboard for that server.
- The **Remote Servers** menu expands to display a list of servers that are monitored by a remote agent. Select a server from the list to access the **Server Analysis** dashboard for the server.
- The **Databases** menu expands to display a list of databases. Select a database from the list to access the **Database Analysis** dashboard for the database.
- The **Dashboards** menu expands to display a list of the dashboards that are available at the global level, or for the current agent, server or database. Select a dashboard from the list to navigate to that dashboard.

## Creating custom charts and dashboards

PEM (version 4.0 and above) allows you to create your own **Charts** and **Dashboards**, allowing you to tailor the interface to the requirements of your organization or individual responsibility.

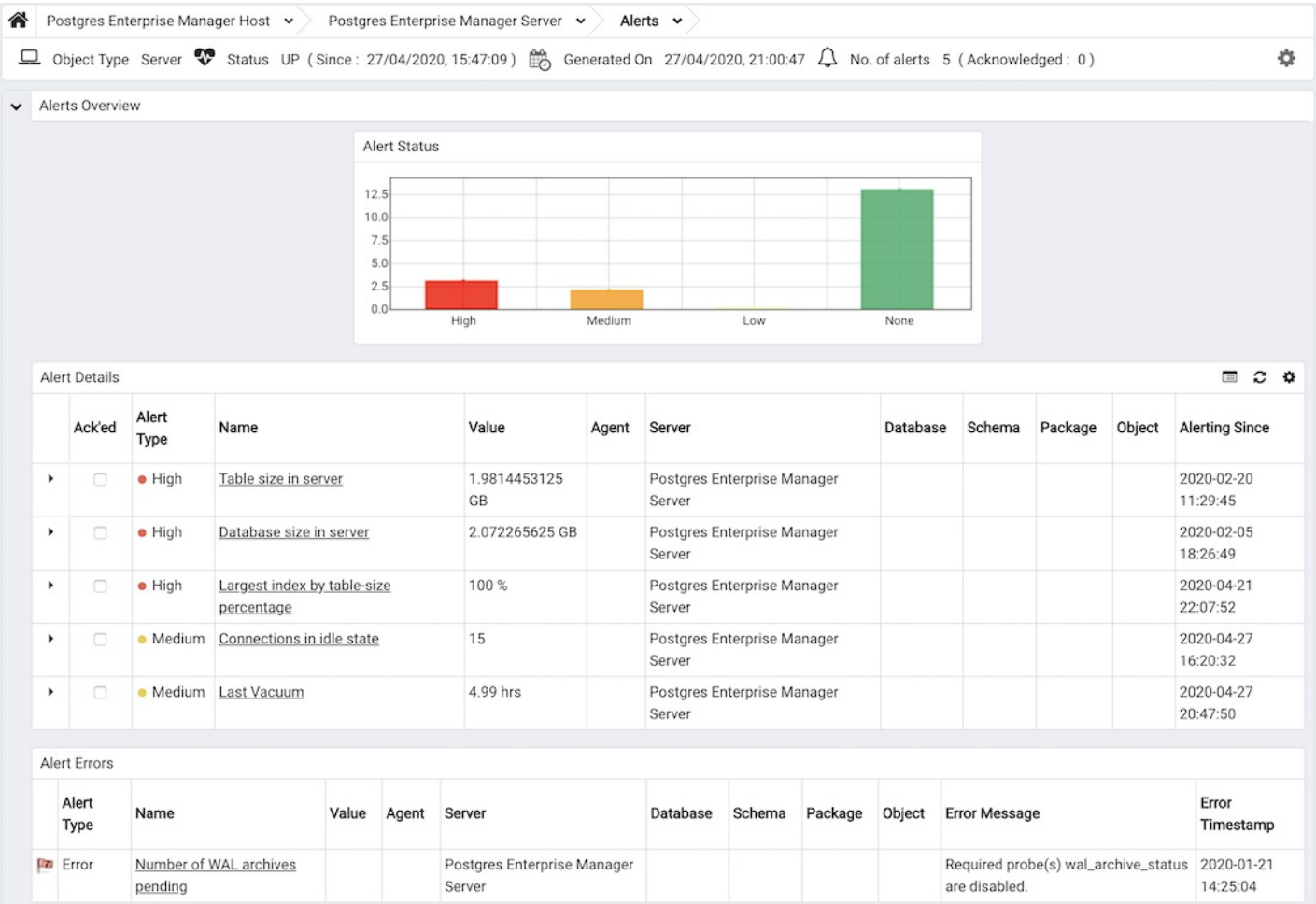
## Available Dashboards

PEM offers the following dashboards:

- **Alerts Dashboard**
- **Audit Log Dashboard**
- **PGD (EDB Postgres Distributed) Monitoring Dashboard**
- **Database Analysis Dashboard**
- **Global Overview Dashboard**
- **I/O Analysis Dashboard**
- **Memory Analysis Dashboard**
- **Object Activity Analysis Dashboard**
- **Operating System Analysis Dashboard**
- **Probe Log Analysis Dashboard**
- **Server Analysis Dashboard**
- **Server Log Analysis Dashboard**
- **Session Activity Analysis Dashboard**
- **Session Wait Analysis Dashboard**
- **Storage Analysis Dashboard**
- **System Wait Analysis Dashboard**
- **Streaming Replication Analysis Dashboard**

34.4.1.1 The Alerts Dashboard

The Alerts Dashboard displays the currently triggered alerts; if opened from the Global Overview, the dashboard displays the current alerts for all monitored nodes on the system. If the Alerts Dashboard is opened from a node within a monitored hierarchy, the report will reflect alerts related to that node, and all monitored objects that reside below that object in the tree control.



Use parameters on the PEM Server Configurations dialog to specify the auto-refresh rate for the Alerts dashboard. To access the Server Configuration dialog, select Server Configuration... from the PEM web interface Management menu.

The Alerts Dashboard header includes the date and time that the page was last updated and a current count of triggered alerts.

The Alerts Overview provides an overview of triggered alerts. The right-most bar indicates the total number of configured alerts that are not currently in an alert state; the three left-most bars indicate the number of Low, Medium and High alerts for the selected object. The vertical key on the left side of the graph provides an alert count.

The Alert Details table lists the currently triggered alerts for the selected object; if opened from the global overview, the Alert Details table lists all of the currently triggered alerts for all monitored objects. Click a column heading to sort the table by the contents of a selected column; click a second time to reverse the sort order. The table contains detailed information about each alert:

- An alert level icon displays in red for a High severity alert, in orange for a Medium severity alert, and in yellow for a Low severity alert.
- Use the arrow to the right of the alert level icon to access a dialog with detailed information about the alert. Within the dialog, the Details tab displays detailed information about the condition that triggered the alert; the Parameters tab displays the values of parameters used in the alert definition. Not all alerts return data that can be viewed on the Details dialog; for information about which templates display detailed metrics, please see the alert templates list



Alert Details (Auto-refresh paused whilst rows are expanded. ⓘ)

| Ack'd                    | Alert Type | Name                                 | Value           | Agent | Server                             | Database | Schema | Package | Object | Alerting Since      |
|--------------------------|------------|--------------------------------------|-----------------|-------|------------------------------------|----------|--------|---------|--------|---------------------|
| <input type="checkbox"/> | ● High     | <a href="#">Table size in server</a> | 1.9814453125 GB |       | Postgres Enterprise Manager Server |          |        |         |        | 2020-02-20 11:29:45 |

General

Parameters

| Table name       | Schema name | Database name | Total table size(MB) |
|------------------|-------------|---------------|----------------------|
| table_statistics | pemhistory  | pem           | 1087                 |
| server_logs      | pemdata     | pem           | 263                  |
| index_statistics | pemhistory  | pem           | 237                  |
| session_info     | pemhistory  | pem           | 137                  |
| lock_info        | pemhistory  | pem           | 88                   |

- The **Ack'd** column provides a checkbox to allow you to acknowledge an alert to prevent additional notifications being sent. This flag is cleared automatically if the alert condition clears and is then detected again.
- The **Alert Type** column indicates the severity of the alert.
- The **Name** column displays the names of the currently triggered alerts. Click the name of an alert to open the **Alerting** configuration dialogue that defines the alert.
- The **Value** column displays the value of the metric that triggered the alert.
- If applicable, the **Agent** column indicates the name of the agent on which the alert is defined.
- If applicable, the **Server** column indicates the name of the server triggering the error message.
- If applicable, the **Database** column indicates the name of the database on which the alert is defined.
- If applicable, the **Schema** column indicates the name of the schema on which the alert is defined.
- If applicable, the **Package** column indicates the name of the package on which the alert is defined.
- If applicable, the **Object** column indicates the name of the monitored object on which the alert is defined.
- If the alert definition includes specified parameters, the parameter values are displayed in the **Additional Params** column.
- If the alert definition includes additional specified parameters, the additional parameter values are displayed in the **Additional Params Value** column.
- The **Alerting Since** column displays the date and time that the alert triggered.

The **Alert Errors** table displays configuration-related errors (eg. accidentally disabling a required probe, or improperly configuring an alert parameter):

Alert Errors

| Alert Type | Name                           | Value | Agent | Server                             | Database | Schema | Package | Object | Error Message                                      | Error Timestamp     |
|------------|--------------------------------|-------|-------|------------------------------------|----------|--------|---------|--------|--|---------------------|
| Error      | Number of WAL archives pending |       |       | Postgres Enterprise Manager Server |          |        |         |        | Required probe(s) wal_archive_status are disabled. | 2020-01-21 14:25:04 |

- An alert indicator in the left-most column indicates that the alert was triggered by an Error.
- The **Alert Type** column indicates the severity of the alert.
- The **Name** column displays the name of the alert. Click an alert name to open the configuration dialogue for the alert.
- The **Value** column displays the value of the metric that triggered the alert, if applicable.
- If applicable, the **Agent** column displays the name of the agent triggering the alert.
- If applicable, the **Server** column displays the name of the server triggering the alert.
- If applicable, the **Database** column indicates the name of the database on which the alert is defined.
- If applicable, the **Schema** column indicates the name of the schema on which the alert is defined.
- If applicable, the **Package** column indicates the name of the package on which the alert is defined.
- If applicable, the **Object** column indicates the name of the monitored object on which the alert is defined.
- The **Error Message** column describes the condition that triggered the alert.
- The **Error Timestamp** column displays the date and time that the alert was triggered.

## Customizing the Alerts Dashboard

You can customize tables and charts that appear on the Alerts dashboard. To open the **Personalize chart configuration** dialog, click the wrench icon in the upper-right corner.

Personalize Chart Configuration

Auto refresh

(10 seconds - 7200 seconds)

60

Download format

JPEG

Colors

High

Medium

Low

None

The fields displayed on the dialog will vary based on the table or chart from which the dialog is opened.

Personalize Chart Configuration

Auto refresh

(10 seconds - 7200 seconds)

60

Show acknowledged alerts?

Yes

|   | Alert | Name                                   | Value           | Agent | Server                             | Database | Schema | Package | Object | Alerting Since      |
|---|-------|--|-----------------|-------|------------------------------------|----------|--------|---------|--------|---------------------|
| + | High  | Table size in server                   | 1.6845703125 GB |       | Postgres Enterprise Manager Server |          |        |         |        | 2020-02-20 11:29:45 |
| + | High  | Database size in server                | 1.7566015625 GB |       | Postgres Enterprise Manager Server |          |        |         |        | 2020-02-05 18:26:49 |
| + | High  | Connections in idle state              | 21              |       | Postgres Enterprise Manager Server |          |        |         |        | 2020-04-27 13:58:27 |
| + | High  | Last Vacuum                            | Never ran       |       | Postgres Enterprise Manager Server |          |        |         |        | 2020-01-21 14:26:04 |
| + | High  | Largest index by table size percentage | 100 %           |       | Postgres Enterprise Manager Server |          |        |         |        | 2020-04-21 22:07:52 |

Use fields on the **Personalize chart configuration** dialog to provide your display preferences:

- Use the **Auto Refresh** field to specify the number of seconds between updates of the data displayed in the table or chart.
- If applicable, use the **Download as** field to indicate if you would like a chart to be downloaded as a JPEG image or a PNG image.
- If applicable, use the **Colours** selectors to specify the display colors that will be used on a chart.
- If applicable, set the **Show Acknowledged Alerts** switch to **Yes** indicate that you would like the table to display alerts that you have acknowledged with a checkbox in the **Ack'ed** column. Set the field to **No** to indicate that the table should hide any acknowledged alerts. The switch acts as a toggle; acknowledged alerts are not purged from the table content until the time specified in the alert definition passes.

To save your customizations, click the save icon (a check mark) in the upper-right corner; to delete any previous changes and revert to the default values, click the delete icon. Save and Delete drop-down menus allow you to specify if your preferences should be applied to **All Dashboards**, or to a selected server or database. Use the close icon to close the **Personalize chart configuration** dialog without preserving your changes.

### 34.4.1.2 The Audit Log Analysis Dashboard

The Audit Log Dashboard allows you to browse the audit logs that have been collected from Advanced Server instances which have enabled audit logging and collection with the **Audit Manager**. If the Audit Log Dashboard is opened from the Global level, it will display logs from all servers. If opened from the Agent level, it will show logs from all servers monitored by that Agent. If opened from the Server level, it will show logs from that server only.

Properties

SQL

Statistics

Dependencies

Dependents

Monitoring

Audit Log

Postgres Enterprise Manager Host

Audit Log

Object Type

Host Agent

Status

UP ( Since : 15/04/2020, 19:35:06 )

Generated On

29/04/2020, 10:08:32

No. of alerts

None

Audit Logs

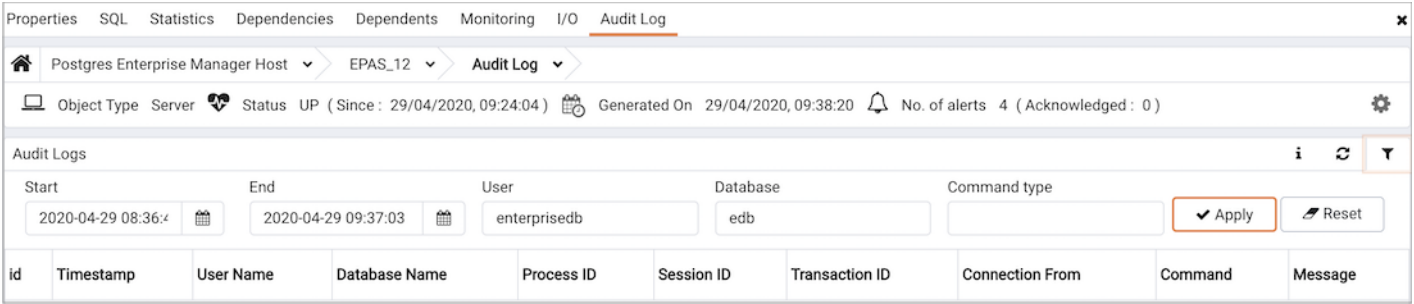
| id   | Server  | Timestamp            | User Name    | Database Name | Process ID | Session ID    | Transaction ID | Connection From | Command        | Message  |
|------|---------|----------------------|--------------|---------------|------------|---------------|----------------|-----------------|----------------|--|
| 7879 | EPAS_12 | 29/04/2020, 10:00:07 | enterprisedb | postgres      | 26391      | 5ea902cf.6717 | 0              | 127.0.0.1:46780 | idle           | disconnection: session time: 0:00:00.014 user=enterprisedb database=postgres host=127.0.0.1 port=46780 |
| 7878 | EPAS_12 | 29/04/2020, 10:00:07 | enterprisedb | postgres      | 26391      | 5ea902cf.6717 | 0              | 127.0.0.1:46780 | idle           | statement: SELECT setting FROM pg_settings WHERE name = 'edb_audit_rotation_seconds'                   |
| 7877 | EPAS_12 | 29/04/2020, 10:00:07 | enterprisedb | postgres      | 26391      | 5ea902cf.6717 | 0              | 127.0.0.1:46780 | idle           | statement: SELECT version();   |
| 7876 | EPAS_12 | 29/04/2020, 10:00:07 | enterprisedb | postgres      | 26391      | 5ea902cf.6717 | 0              | 127.0.0.1:46780 | authentication | connection authorized: user=enterprisedb database=postgres   |
| 7875 | EPAS_12 | 29/04/2020, 10:00:07 | enterprisedb | postgres      | 26388      | 5ea902cf.6714 | 0              | 127.0.0.1:46774 | idle           | disconnection: session time: 0:00:00.007 user=enterprisedb database=postgres host=127.0.0.1 port=46774 |
| 7874 | EPAS_12 | 29/04/2020, 10:00:07 | enterprisedb | postgres      | 26388      | 5ea902cf.6714 | 0              | 127.0.0.1:46774 | authentication | connection authorized: user=enterprisedb database=postgres   |
| 7873 | EPAS_12 | 29/04/2020, 10:00:07 | enterprisedb | postgres      | 26384      | 5ea902cf.6710 | 0              | 127.0.0.1:46766 | idle           | disconnection: session time: 0:00:00.029 user=enterprisedb database=postgres host=127.0.0.1 port=46766 |
| 7872 | EPAS_12 | 29/04/2020, 10:00:07 | enterprisedb | postgres      | 26384      | 5ea902cf.6710 | 0              | 127.0.0.1:46766 | idle           | statement: SELECT setting FROM pg_settings WHERE name='log_temp_files'                                 |
| 7871 | EPAS_12 | 29/04/2020, 10:00:07 | enterprisedb | postgres      | 26384      | 5ea902cf.6710 | 0              | 127.0.0.1:46766 | idle           | statement: SELECT setting FROM pg_settings WHERE name='log_autovacuum_min_duration'                    |
| 7870 | EPAS_12 | 29/04/2020, 10:00:07 | enterprisedb | postgres      | 26384      | 5ea902cf.6710 | 0              | 127.0.0.1:46766 | idle           | statement: SELECT setting FROM pg_settings WHERE name='log_min_duration_statement'                     |
| 7869 | EPAS_12 | 29/04/2020, 10:00:07 | enterprisedb | postgres      | 26384      | 5ea902cf.6710 | 0              | 127.0.0.1:46766 | idle           | statement: SELECT (setting::int/(24*60))::int FROM pg_settings WHERE name = 'log_rotation_age'         |
| 7868 | EPAS_12 | 29/04/2020, 10:00:07 | enterprisedb | postgres      | 26384      | 5ea902cf.6710 | 0              | 127.0.0.1:46766 | idle           | statement: SELECT (setting::int/1024)::int FROM pg_settings WHERE name = 'log_rotation_size'           |
| 7867 | EPAS_12 | 29/04/2020, 10:00:07 | enterprisedb | postgres      | 26384      | 5ea902cf.6710 | 0              | 127.0.0.1:46766 | idle           | statement: SELECT upper(setting) FROM pg_settings WHERE name='syslog_facility';                        |

The **Audit Log Dashboard** header includes the date and time that the page was last updated, and a current count of triggered alerts.

Audit Log table entries are loaded on demand in batches; to load additional entries, scroll to the end of the log and the additional rows will be automatically loaded from the database and added to the table. Log entries are show in chronological order, most recent first.

- The **Id** column identifies the PEM agent that monitors the server that initiated the recorded transaction.
- The **Server** column identifies the server that initiated the recorded transaction.
- The **Timestamp** column shows the date and time that the log entry was made.
- The **User Name** column shows the user which executed the statment in the audit log entry.
- the **Database Name** column shows the database on which the statment in the audit log entry was executed.
- The **Process ID** column shows the ID of the process which executed the statement in the audit log entry.
- The **Session ID** column shows the ID of the session in which the statement in the audit log entry was executed.
- The **Transaction ID** column shows the ID of the transaction in which the statement in the audit log entry was executed.
- The **Connection From** column shows the client's address from where the session was connected.
- The **Command** column shows the type of command executed.
- The **Message** column shows the message associated with the audit log entry.

Click **Show Filters** to display a panel that you can use to filter the audit log entries that are shown in the table below; click on **Hide Filters** to close the panel.



Use the fields within the filter definition box to describe a selection criteria that PEM will use to select a subset of a report for display:

- Use the date and time selectors in the **From** field to specify a starting date and time for the displayed log entries.
- Use the date and time selectors in the **To** field to specify an ending date and time for the displayed log entries.
- Enter a username in the **Username** field to show log entries for the specified user only.
- Enter a database name in the **Database** field to show log entries for the specified database only.
- Enter a command type (for example; 'SELECT', 'authentication' or 'idle') in the **Command Type** field to show log entries of that type only.

### 34.4.1.3 PGD Monitoring Dashboards

The **PGD Monitoring Dashboards** display the information about replication activities of the **PGD Nodes** in a **PGD Group** . There are three different dashboards for PGD:

- PGD Admin Dashboard
- PGD Group Monitoring Dashboard
- PGD Node Monitoring Dashboard

To view the details on these dashboards, you must enable the probes on the PGD Nodes. By default, the **PGD Probes** are disabled. To enable the probes, highlight the name of the PGD Node in the PEM client browser tree control, and select **Manage Probes...** from the **Management** menu. Use the **Manage Probes..** tab to enable to Probes.

#### PGD Admin Dashboard

To view the details on **PGD Admin Dashboard** enable the following probes:

- PGD Group Camo Details
- PGD Group Locks
- PGD Group Raft Details
- PGD Group Versions Details
- PGD Node Summary
- PGD Worker
- PGD Worker Errors

Then highlight the **PGD Node** in the PEM client browser and select the **PGD Admin** from the **Dashboards** drop down menu.

unwrap

bdrdb

PGD Admin

Object Type

Database

Status

UP ( Since : 06/09/2022, 16:04:44 )

Generated On 09/09/2022, 12:49:47

No. of alerts

None

Node Summary

Node Summary

| Node   | Node Group | Peer State | Peer Target State | Sub Repset |
|--------|------------|------------|-------------------|------------|
| unwrap | bdrgroup   | ● ACTIVE   | ACTIVE            |            |
| quirk  | bdrgroup   | ● ACTIVE   | ACTIVE            |            |
| krone  | bdrgroup   | ● ACTIVE   | ACTIVE            |            |

Global Locks

Global Locks

| Origin Node Name                                    | Lock Type | Relation | PID | Acquire Stage | Waiters | Global Lock Request Time | Local Lock Request Time | Last State Change Time |
|---|-----------|----------|-----|---------------|---------|--------------------------|-------------------------|------------------------|
| Not enough data is available to generate the table. |           |          |     |               |         |                          |                         |                        |

EDB Postgres Distributed Group Version Details

PGD Group Version Details

| Node Name | Postgres Version          | pglogical Version | PGD Version | PGD Edition |
|-----------|---------------------------|-------------------|-------------|-------------|
| krone     | 13.8 (2ndQPG 13.8r1.1.10) | N/A               | 3.7.17      | EE          |
| quirk     | 13.8 (2ndQPG 13.8r1.1.10) | N/A               | 3.7.17      | EE          |
| unwrap    | 13.8 (2ndQPG 13.8r1.1.10) | N/A               | 3.7.17      | EE          |

EDB Postgres Distributed Workers

PGD Workers

|   | Worker PID | Worker Role Name | Query Start                | Worker Commit Timestamp    | Worker Wait Event Type |
|---|------------|------------------|----------------------------|----------------------------|------------------------|
| ▶ | 2567       | receiver         |                            | 2022-09-09 07:19:44.446955 | Activity               |
| ▶ | 2568       | receiver         |                            | 2022-09-09 07:19:44.44401  | Activity               |
| ▶ | 2569       | writer           | 2022-09-09 05:34:33.372576 |                            | IPC                    |
| ▶ | 2570       | writer           | 2022-09-09 05:34:33.375272 |                            | IPC                    |

PGD Worker Errors

|                        | Worker PID | Origin Name | Error Message | Error Time | Error Age |
|------------------------|------------|-------------|---------------|------------|-----------|
| No workers were found. |            |             |               |            |           |

EDB Postgres Distributed Group Camo Details

PGD Group Camo Details

| Node Name   | Camo Partner Of | Camo Origin For | Is Camo Partner Connected? | Is Camo Partner Ready? | Camo Transactions Resolved | Apply LSN | Receive LSN | Apply Queue Size |
|---|-----------------|-----------------|----------------------------|------------------------|----------------------------|-----------|-------------|------------------|
| Not enough data is available to generate the table. |                 |                 |                            |                        |                            |           |             |                  |

EDB Postgres Distributed Group Raft Details

PGD Group Raft Details

| Node Name | State         | Leader ID | Current Term | Commit Index |
|-----------|---------------|-----------|--------------|--------------|
| krone     | RAFT_LEADER   | 844970090 | 11           | 407871       |
| quirk     | RAFT_FOLLOWER | 844970090 | 11           | 407871       |
| unwrap    | RAFT_FOLLOWER | 844970090 | 11           | 407871       |

The **Node Summary** table provides information about all the nodes of PGD Group:

- The **Node** column displays the name of the PGD Node in the PGD Group.
- The **Node Group** column displays the name of the PGD Group.
- The **Peer State** column displays the status of the particular PGD Node.
- The **Peer Target State** column displays the status of the other node which is tried to reach.
- The **Sub Repset** column displays the name of the subscribed replication sets.

The **Global Locks** table provides information about PGD's global locking activity and progress of PGD Group:

- The **Origin Node Name** column displays the name of the node from where the global lock is originated.
- The **Lock Type** column displays the type of the Lock. The possible values are **GLOBAL\_LOCK\_DDL** or **GLOBAL\_LOCK\_DML**.
- The **relation** column displays the name of the locked relation(for DML Locks) or keys(for advisory locks).
- The **PID** column displays the PID of the process holding the lock.
- The **Acquire Stage** column displays the internal state of the lock acquisition process.
- The **Waiters** column displays the list of the backends waiting for the same global lock.

- The **Global Lock Request Time** column displays the timestamp at which the process of acquiring the global lock started. It can be **Null** if time is unknown on the current node.
- The **Local Lock Request Time** column displays the timestamp at which the local node tried to acquire the local lock for this global lock. This includes the time taken for the heavyweight session lock acquire, but does NOT include any time taken on DDL epochs or queue flushing. If the lock is re-acquired after local node restart, this will be the node restart time.
- The **Last State Change Time** column displays the timestamp at which the **Acquire Stage** last changed.

The **PGD Group Version Details** table provides version information of the installed Postgres, pglogical, PGD, and PGD edition for each node:

- The **Node Name** column displays the name of PGD Node in the PGD Group.
- The **Postgres Version** column displays the version of the installed postgres on each PGD Node.
- The **pglogical Version** column displays the version of the installed pglogical on each PGD Node.
- The **PGD Version** column displays the version of the installed PGD extension on each PGD Node.
- The **PGD Edition** column displays the Edition of the installed PGD extension on each PGD Node. The possible values are **EE** or **SE**.

The **PGD Workers** table provides information about the **PGD background worker processes**:

- The **Worker PID** column displays the process id of the worker process.
- The **Worker Role Name** column displays the role name of the worker processes. The possible values are **receiver** or **writer**.
- The **Query Start** column displays the query start timestamp when a particular query started executing.
- The **Worker Commit Timestamp** column displays the timestamp at which the worker process committed the transaction.
- The **Worker Wait Event Type** column displays the type of wait event for the worker process.

The **PGD Worker Errors** table provides information about the persistent log of errors from the PGD background worker processes.

- The **Worker PID** column displays the process id of the worker process.
- The **Origin Name** column displays the name of the PGD Node that initiated the error from the worker process.
- The **Error Message** column displays the description of the error from the worker process.
- The **Error Time** column displays the timestamp of the error from the worker process.
- The **Error Age** column displays the time elapsed in years, months and days since the error occurred.

The **PGD Group Camo Details** table provides information relate to CAMO from all the PGD Nodes.

- The **Node Name** column displays the name of PGD Node in the PGD Group.
- The **Camo Partner of** column displays the node name for whom this node is partner.
- The **Camo Origin For** column displays the node name for whom this node is origin.
- The **Is Camo Partner Connected?** column displays the connection status of the camo partner.
- The **Is Camo Partner Ready?** column displays the readiness status of the camo partner.
- The **Camo Transactions Resolved** column displays if there are any pending and unresolved CAMO transactions.
- The **Apply LSN** column displays the latest position reported as replayed.
- The **Receive LSN** column displays the latest LSN of any change or message received.
- The **Apply Queue Size** column displays the difference between apply\_lsn and receive\_lsn in bytes.

The **PGD Group Raft Details** table provides detail information about the PGD Group Raft:

- The **Node Name** column displays the name of PGD Node in the PGD Group.
- The **State** column displays the Raft worker state on the PGD Node.
- The **Leader ID** column displays the Node id of the RAFT\_LEADER.
- The **Current Term** column displays the Raft election internal id.
- The **Commit Index** column displays the Raft snapshot internal id.

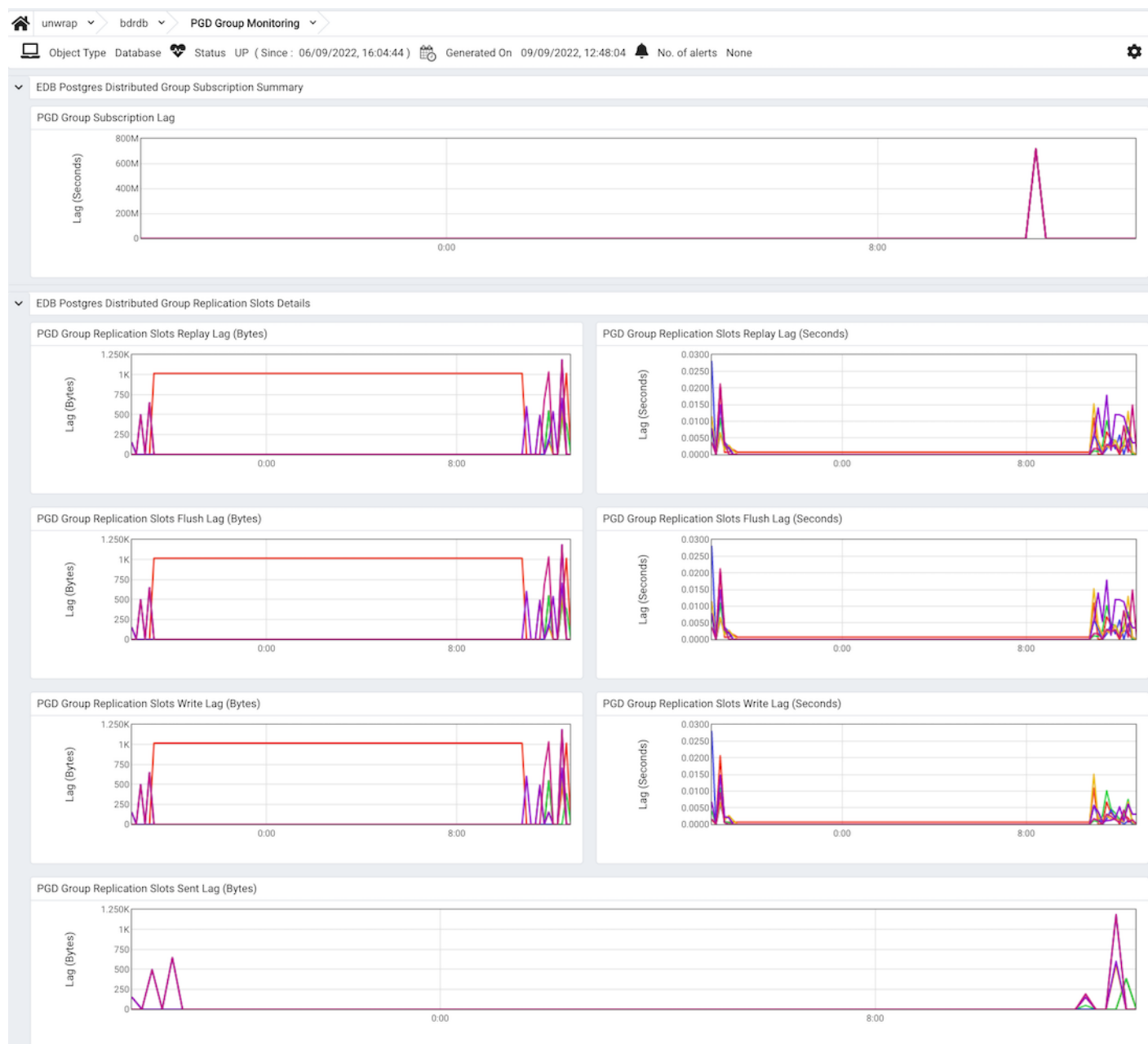
## PGD Group Monitoring Dashboard

To view the details on the **PGD Group Monitoring** dashboard enable the following probes:



- PGD Group Replication Slots Details
- PGD Group Subscription Summary

Then highlight the **PGD Node** in the PEM client browser and select the **PGD Group Monitoring** from the **Dashboards** drop down menu.



- The **PGD Group Subscription Lag** graph displays the delay to apply the transactions on subscribed nodes as compared to the origin. The delay in applying transactions between the subscribed node and origin is displayed with different colors on the graph. The vertical key on the left side of the graph indicates the delay to apply the transactions on subscribed nodes in seconds as compared to the origin.
- The **PGD Group Replication Slots Replay Lag(Bytes)** graph displays the difference between the replay\_lsn and current WAL write position in bytes. The difference among the different set of replication slots is displayed with different color on the graph. The vertical key on the left side of the graph indicates the difference in bytes.
- The **PGD Group Replication Slots Replay Lag(seconds)** graph displays the approximate lag time for reported replay among the replication slots. The delay time between the subscribed replication slots is displayed with different colors on the graph. The vertical key on the left side of the graph indicates the lag time in seconds.
- The **PGD Group Replication Slots Flush Lag(Bytes)** graph displays the difference between the flush\_lsn and current WAL write position in bytes. The difference among the different set of replication slots is displayed with different colors on the graph. The vertical key on the left side of the graph indicates the difference in bytes.
- The **PGD Group Replication Slots Flush Lag(seconds)** graph displays the approximate lag time for reported flush among the replication slots. The delay time between the subscribed replication slots is displayed with different colors on the graph. The vertical key on the left side of the graph indicates the lag time in seconds.
- The **PGD Group Replication Slots Write Lag(Bytes)** graph displays the difference between the write\_lsn and current WAL write

position in bytes. The difference among the different set of subscribed replication slots is displayed with different colors on the graph. The vertical key on the left side of the graph indicates the difference in bytes.

- The **PGD Group Replication Slots Write Lag(seconds)** graph displays the approximate lag time for reported write among the replication slots. The delay time between the subscribed replication slots is displayed with different colors on the graph. The vertical key on the left side of the graph indicates the lag time in seconds.
- The **PGD Group Replication Slots Sent Lag(Bytes)** graph displays the difference between sent\_lsn and WAL write position in bytes. The difference among the different set of subscribed replication slots is displayed with different colors on the graph. The vertical key on the left side of the graph indicates the difference in bytes.

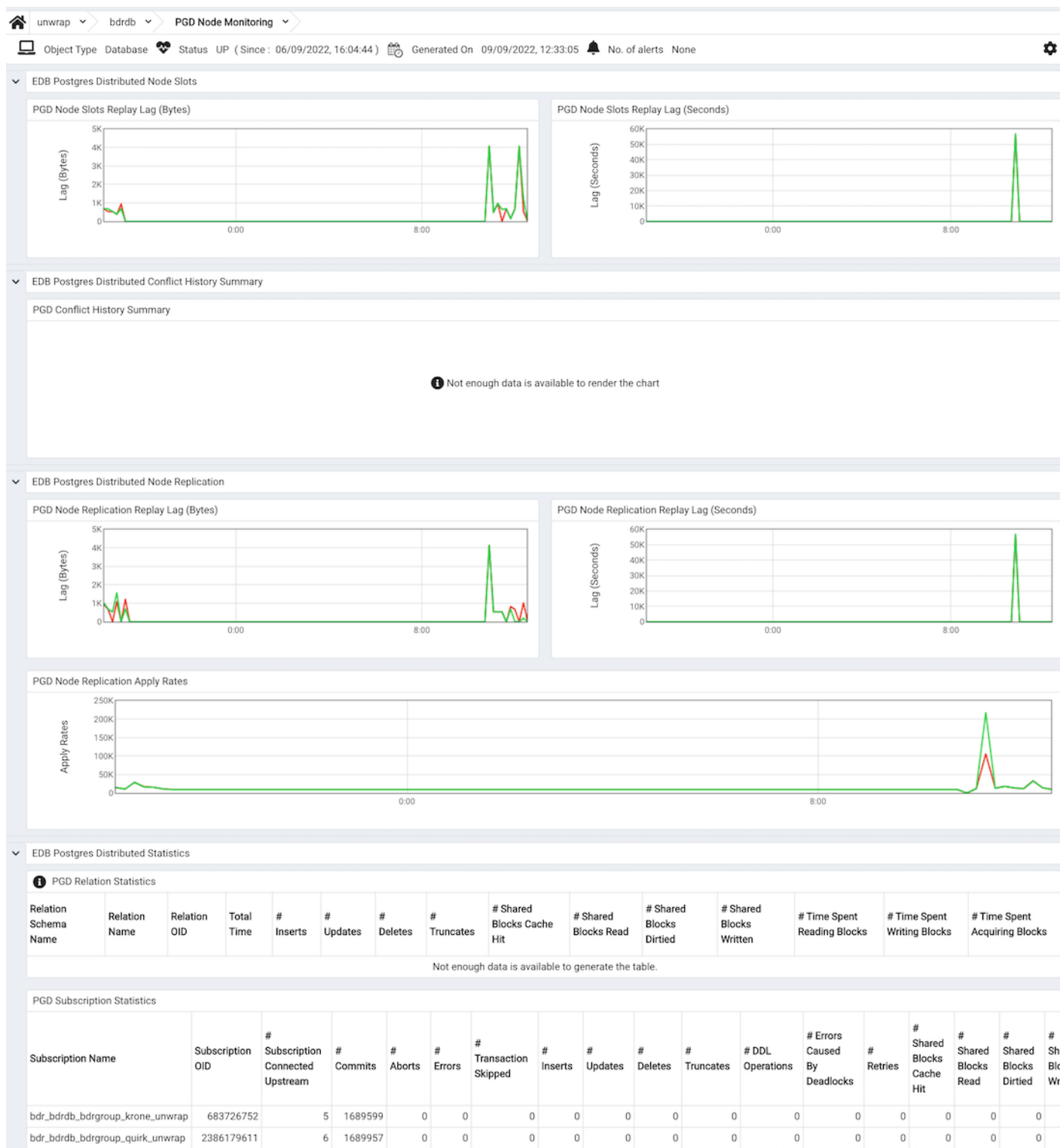
## PGD Node Monitoring Dashboard

To view the details on the **PGD Node Monitoring** dashboard enable the following probes:

- PGD Conflict History Summary
- PGD Node Replicate Rates
- PGD Node Slots

Then highlight the **PGD Node** in the PEM client browser and select the **PGD Node Monitoring** from the **Dashboards** drop down menu.

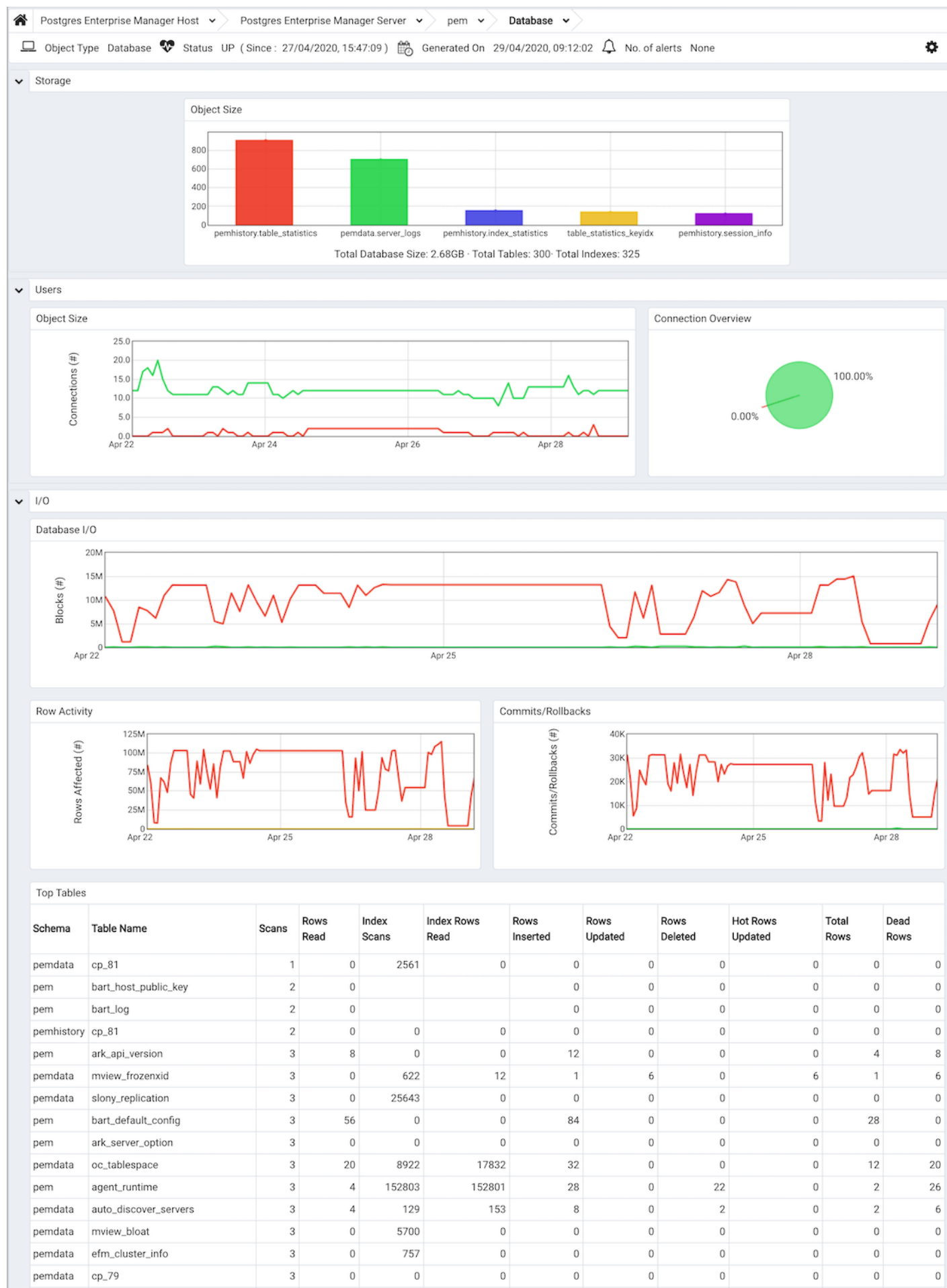




- The **PGD Node Slots Replay Lag(Bytes)** graph displays the difference between the replay\_lsn and current WAL write position in bytes for the particular PGD Node. The vertical key on the left side of the graph indicates the difference in bytes.
- The **PGD Conflict History Summary** graph displays the summary on the conflict history per conflict type for the particular PGD Node. The vertical key on the left side of the graph indicates the number of row conflicts.
- The **PGD Node Replication Replay Lag(Bytes)** graph displays the difference between the replay\_lsn and current WAL write position on origin in bytes for the particular PGD Node. The vertical key on the left side of the graph indicates the difference in bytes.
- The **PGD Group Replication Slots Replay Lag(seconds)** graph displays the approximate lag time for reported replay among the replication slots for the particular PGD Node. The vertical key on the left side of the graph indicates the lag time in seconds.
- The **PGD Node Replication Apply Rates** graph displays the information about the outgoing replication activity from a given PGD Node. The vertical key on the left side of the graph.

### 34.4.1.4 The Database Analysis Dashboard

The Database Analysis dashboard provides a high-level overview of database activity for the selected database, including a comparative storage analysis of the 5 largest tables/indexes, user activity analysis, weekly I/O analysis, and an activity analysis of the tables that reside in the selected database.



|         |                                    |   |    |        |       |    |   |   |   |    |    |
|---------|------------------------------------|---|----|--------|-------|----|---|---|---|----|----|
| pem     | tbl_chart                          | 3 | 26 | 108318 | 278   | 40 | 0 | 1 | 0 | 13 | 27 |
| pem     | server_version                     | 3 | 50 | 574    | 830   | 75 | 0 | 0 | 0 | 25 | 50 |
| pem     | dashboard_settings                 | 3 | 0  | 173990 | 0     | 0  | 0 | 0 | 0 | 0  | 0  |
| pemdata | streaming_replication_db_conflicts | 3 | 0  | 0      | 0     | 0  | 0 | 0 | 0 | 0  | 0  |
| pemdata | number_of_prepared_transactions    | 3 | 10 | 56941  | 56482 | 16 | 0 | 0 | 0 | 6  | 10 |
| pemdata | function_statistics                | 3 | 0  | 5585   | 0     | 0  | 0 | 0 | 0 | 0  | 0  |
| pem     | chart_metric                       | 3 | 46 | 66752  | 89143 | 38 | 0 | 4 | 0 | 32 | 6  |
| pem     | bart_server_config                 | 3 | 0  | 0      | 0     | 0  | 0 | 0 | 0 | 0  | 0  |
| pemdata | cp_80                              | 3 | 0  | 0      | 0     | 0  | 0 | 0 | 0 | 0  | 0  |
| pem     | probe_config_view                  | 3 | 0  | 54     | 0     | 0  | 0 | 0 | 0 | 0  | 0  |

Use parameters on the [PEM Server Configurations](#) dialog to specify the auto-refresh rate for the dashboard. To access the [Server Configuration](#) dialog, select [Server Configuration...](#) from the PEM web interface [Management](#) menu.

The Database Analysis dashboard header displays the date and time that the server started, the date and time that the Database Analysis dashboard was last updated, and the number of alerts currently triggered for the specified database (and monitored objects that reside within that database).

The [Storage](#) bar graph plots the relative size of the 5 largest tables and indexes that reside within the selected database. The vertical key on the left side of the graph indicates each table or index in megabytes; the key on the right side of the chart identifies the tables and indexes by name.

The [Users](#) section of the Database Analysis dashboard displays information about user connections:

- The [User Activity](#) graph plots the active and idle connections over the previous week. The vertical key on the left side of the chart indicates the connection count.
- The [Connection Overview](#) chart provides a comparative display of the active and idle connections currently established with the server (when the most recent probe executed).

The graphs in the [I/O](#) section present an analysis of I/O activity over the previous week.



- The [Database I/O](#) graph plots the number of blocks found in cached memory and the number of blocks read from disk over the previous week. The vertical key on the left side of the graph indicates number of blocks hit.
- The [Row Activity](#) graph displays the row activity for tables residing within the database over the previous week.

The vertical key on the left side of the graph indicates the number of rows.

- The [Commits/Rollbacks](#) graph displays the number of transactions committed and rolled back within the selected database over the previous week.

The vertical key on the left side of the graph indicates the transaction count.

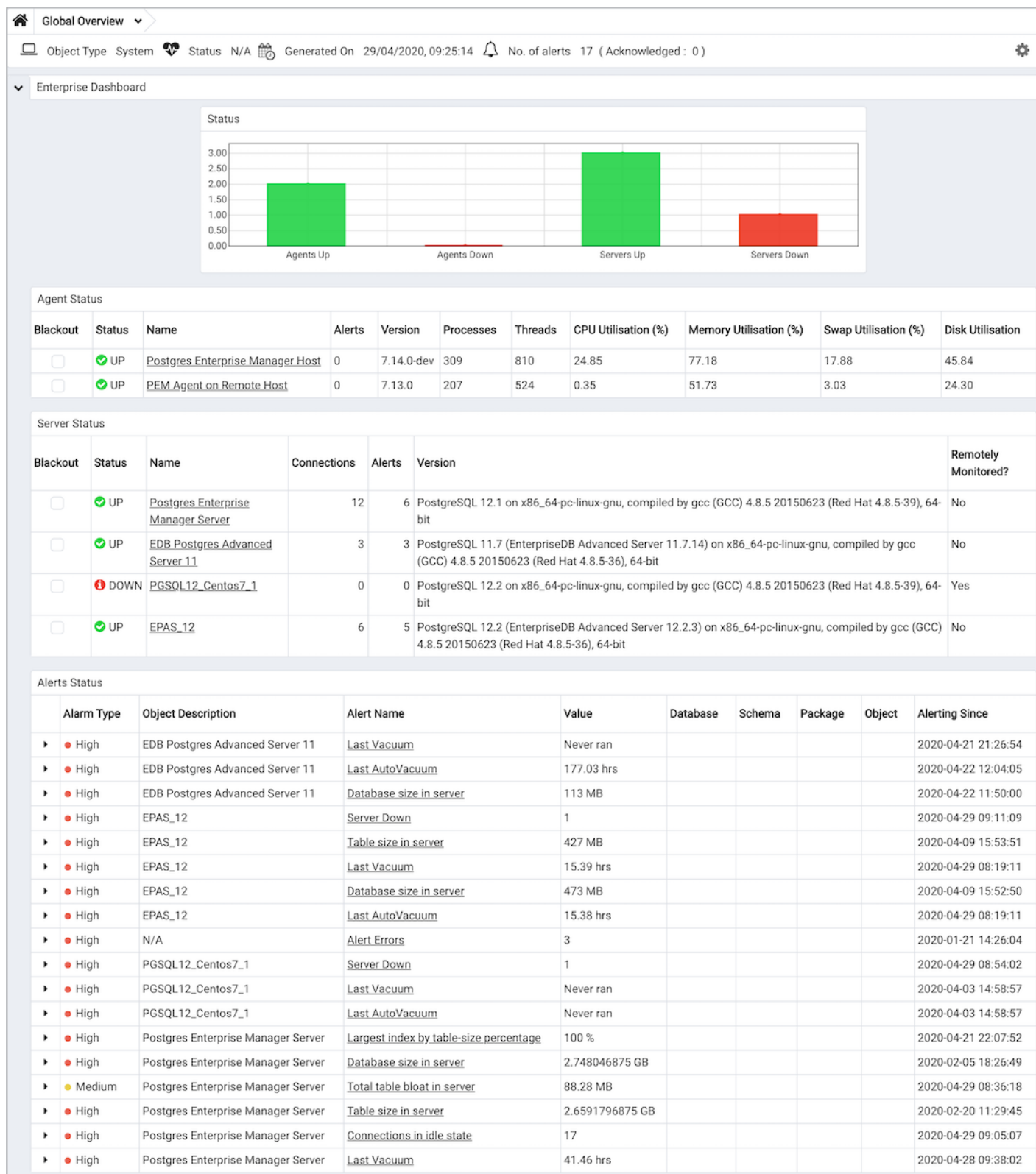
| Top Tables |                                    |       |           |             |                 |               |              |              |                  |            |           |
|------------|------------------------------------|-------|-----------|-------------|-----------------|---------------|--------------|--------------|------------------|------------|-----------|
| Schema     | Table Name                         | Scans | Rows Read | Index Scans | Index Rows Read | Rows Inserted | Rows Updated | Rows Deleted | Hot Rows Updated | Total Rows | Dead Rows |
| pemdata    | cp_81                              | 1     | 0         | 2561        | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pem        | bart_host_public_key               | 2     | 0         |             |                 | 0             | 0            | 0            | 0                | 0          | 0         |
| pem        | bart_log                           | 2     | 0         |             |                 | 0             | 0            | 0            | 0                | 0          | 0         |
| pemhistory | cp_81                              | 2     | 0         | 0           | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pem        | ark_api_version                    | 3     | 8         | 0           | 0               | 12            | 0            | 0            | 0                | 4          | 8         |
| pemdata    | mview_frozenxid                    | 3     | 0         | 622         | 12              | 1             | 6            | 0            | 6                | 1          | 6         |
| pemdata    | slony_replication                  | 3     | 0         | 25643       | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pem        | bart_default_config                | 3     | 56        | 0           | 0               | 84            | 0            | 0            | 0                | 28         | 0         |
| pem        | ark_server_option                  | 3     | 0         | 0           | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pemdata    | oc_tablespace                      | 3     | 20        | 8922        | 17832           | 32            | 0            | 0            | 0                | 12         | 20        |
| pem        | agent_runtime                      | 3     | 4         | 152803      | 152801          | 28            | 0            | 22           | 0                | 2          | 26        |
| pemdata    | auto_discover_servers              | 3     | 4         | 129         | 153             | 8             | 0            | 2            | 0                | 2          | 6         |
| pemdata    | mview_bloat                        | 3     | 0         | 5700        | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pemdata    | efm_cluster_info                   | 3     | 0         | 757         | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pemdata    | cp_79                              | 3     | 0         | 0           | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pem        | tbl_chart                          | 3     | 26        | 108318      | 278             | 40            | 0            | 1            | 0                | 13         | 27        |
| pem        | server_version                     | 3     | 50        | 574         | 830             | 75            | 0            | 0            | 0                | 25         | 50        |
| pem        | dashboard_settings                 | 3     | 0         | 173990      | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pemdata    | streaming_replication_db_conflicts | 3     | 0         | 0           | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pemdata    | number_of_prepared_transactions    | 3     | 10        | 56941       | 56482           | 16            | 0            | 0            | 0                | 6          | 10        |
| pemdata    | function_statistics                | 3     | 0         | 5585        | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pem        | chart_metric                       | 3     | 46        | 66752       | 89143           | 38            | 0            | 4            | 0                | 32         | 6         |
| pem        | bart_server_config                 | 3     | 0         | 0           | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pemdata    | cp_80                              | 3     | 0         | 0           | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pem        | probe_config_view                  | 3     | 0         | 54          | 0               | 0             | 0            | 0            | 0                | 0          | 0         |

The **Hot Tables** table provides a detailed analysis of the activity for each table that resides within the selected database. Click a column heading to sort the table by the values within the column; click again to reverse the sort order.

- The **Schema** column identifies the schema in which the table resides.
- The **Table Name** column identifies the name of the table.
- The **Scans** column displays the number of scans performed on the table.
- The **Rows Read** column displays the number of rows read from the specified table.
- The **Index Scans** column displays the number of index scans performed on the specified table.
- The **Index Rows Read** column displays the number of rows read during index scans on the specified table.
- The **Rows Inserted** column displays the number of rows inserted into the specified table.
- The **Rows Updated** column displays the number of rows updated in the specified table.
- The **Rows Deleted** column displays the number of rows deleted from the specified table.
- The **Hot Rows Updated** column displays the number of hot row updates into the table; when a hot row update occurs, the new row occupies the same page as the previous row.
- The **Total Rows** column displays the number of total rows in the table.
- The **Dead Rows** column displays the number of rows that have been deleted, but have not been reclaimed via a VACUUM command or the AUTOVACUUM process.

### 34.4.1.5 The Global Overview Dashboard

Upon connecting to Postgres Enterprise Manager, the web interface displays the **Global Overview** dashboard. The Global Overview dashboard displays the status of each PEM server and agent, and calls your attention to any triggered alerts on monitored objects.



Use parameters on the [PEM Server Configurations](#) dialog to specify the auto-refresh rate for the dashboard. To access the [Server Configuration](#) dialog, select [Server Configuration...](#) from the PEM web interface [Management](#) menu.

The [Global Overview](#) header displays the date and time that the overview was last updated and the current number of triggered alerts.

The [Enterprise Dashboard](#) bar graph provides an at-a-glance overview of the status of your PEM agents and servers.

The [Agent Status](#) table provides detailed information about the status of each individual PEM agent:

- Check the box in the [Blackout](#) column to disable alert processing for the agent and all servers monitored by the agent. This is useful when

undertaking maintenance on the agent or the host on which the agent runs.

- The **Status** column reports the current state of the agent; **UP**, **DOWN** or **UNKNOWN**. A healthy agent displays a green 'check' circle icon; an agent that is down displays a red 'info' circle icon; an agent registered with PEM, but - never sent an heartbeat, displays a gray 'question' circle icon. If user changes the colour for UP, DOWN or UNKNOWN status of agents in *\*Enterprise Dashboard\** bar chart, then that color will be reflected for the respective status icon.
- The **Name** column displays the name of the agent. Click the name to navigate to the **Operating System Analysis** dashboard for the selected host.
- The **Alerts** column displays the number of current alerts triggered on the server.
- The **Version** column displays the agent's version.
- The **Processes** column lists the number of processes running on the agent's host.
- The **Threads** column lists the number of threads running on the agent's host.
- The **CPU Utilisation (%)** column shows the average utilisation of all CPU cores on the host.
- The **Memory Utilisation (%)** column shows the percentage of available RAM memory used on the host.
- The **Swap Utilisation (%)** column shows the percentage of available swap memory used on the host.
- The **Disk Utilisation** column shows the total percentage of disk space used, for all disks on the host.

The **Postgres Server Status** table provides detailed information about the status of each individual server:

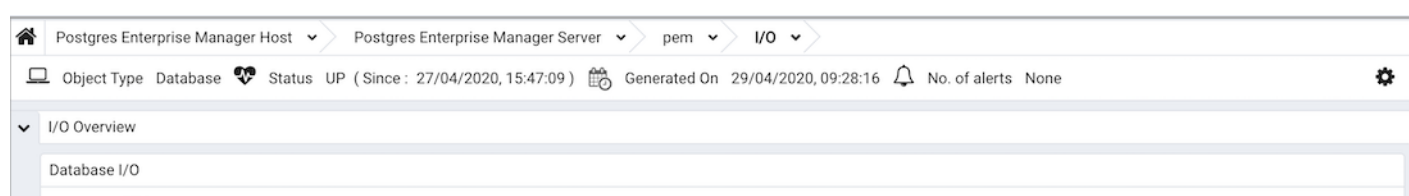
- Check the box in the **Blackout** column provides a checkbox to disable alert processing for the server. This is useful when performing maintenance on the server.
- The **Status** column reports the current state of the server; UP, DOWN, UNKNOWN or UNMANAGED.
- A healthy server displays a green 'check' circle icon; a disabled server displays a red 'info' circle icon; an unknown server displays a grey 'question' circle icon; an unmanaged server displays a light gray 'user' circle icon. If user changes the colour for UP, DOWN, UNKNOWN or UNMANAGED servers in **Enterprise Dashboard** bar chart, then that color will be reflected for the respective status icon.
- The **Name** column displays the name of the agent. Click the name to navigate to the **Operating System Analysis** dashboard for the host.
- The **Connections** column reports the current number of connections to the server.
- The **Alerts** column displays the number of current alerts triggered on the server.
- The **Version** column lists the PostgreSQL version and build signature.
- The **Remotely Monitored** column displays a **Yes** if the PEM agent that is bound to the monitored server does not reside on the same host as the server, and a **No** if the agent resides on the same host as the server.

Triggered alerts displayed in the **Alert Status** table include both PEM-defined alerts and user-defined alerts for all PEM-monitored hosts, servers, agents and database objects. The **Alert Status** table will also display an alert if an **agent or server is down**.

- The **Alarm Type** column reports the alert severity. An icon displays in red for a **High** severity alert, in yellow for a **Medium** severity alert, and in grey for a **Low** severity alert.
- The **Object Description** column displays a description of the object that triggered the alert.
- The **Alert Name** column displays the name of the triggered alert. When viewing the dashboard in the PEM client, you can click the Alert Name to open the configuration dialogue for the alert.
- The **Value** column displays the current value of the object that triggered the alert.
- The **Database** column displays the name of the database with which the alert is associated (if applicable).
- The **Schema** column displays the name of the schema with which the alert is associated (if applicable).
- The **Package** column displays the name of the package with which the alert is associated (if applicable).
- The **Object** column displays the name of the object with which the alert is associated (if applicable).
- The **Additional Params** column displays any additional parameters specified for the alert.
- The **Additional Param Values** column displays any additional parameter values specified for the alert.
- The **Alerting Since** column displays the date and time at which the alert triggered.

### 34.4.1.6 The I/O Analysis Dashboard

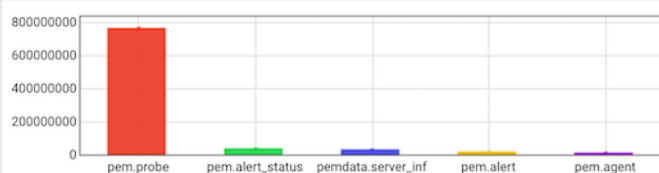
The I/O Analysis dashboard displays usage statistics for a specific database.







▼ Top Tables/Indexes



#### Object I/O Details

### Objects Activity

| Schema     | Table Name                         | Scans | Rows Read | Index Scans | Index Rows Read | Rows Inserted | Rows Updated | Rows Deleted | Hot Rows Updated | Total Rows | Dead Rows |
|------------|------------------------------------|-------|-----------|-------------|-----------------|---------------|--------------|--------------|------------------|------------|-----------|
| permdata   | cp_81                              | 1     | 0         | 2561        | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| permhistry | cp_81                              | 2     | 0         | 0           | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pem        | bart_log                           | 2     | 0         |             |                 |               | 0            | 0            | 0                | 0          | 0         |
| pem        | bart_host_public_key               | 2     | 0         |             |                 |               | 0            | 0            | 0                | 0          | 0         |
| pem        | data_chart                         | 3     | 6         | 257         | 245             | 10            | 0            | 1            | 0                | 3          | 7         |
| pem        | bart_config                        | 3     | 0         | 0           | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| permdata   | cp_80                              | 3     | 0         | 0           | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| permdata   | function_statistics                | 3     | 0         | 5585        | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pem        | bart_server_config                 | 3     | 0         | 0           | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pem        | bart_default_config                | 3     | 56        | 0           | 0               | 84            | 0            | 0            | 0                | 28         | 0         |
| permdata   | number_of_prepared_transactions    | 3     | 10        | 56941       | 56482           | 16            | 0            | 0            | 0                | 6          | 10        |
| pem        | tbl_chart                          | 3     | 26        | 108318      | 278             | 40            | 0            | 1            | 0                | 13         | 27        |
| permdata   | oc_tablespace                      | 3     | 20        | 8922        | 17832           | 32            | 0            | 0            | 0                | 12         | 20        |
| pem        | ark_server_option                  | 3     | 0         | 0           | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pem        | server_version                     | 3     | 50        | 574         | 830             | 75            | 0            | 0            | 0                | 25         | 50        |
| pem        | probe_config_view                  | 3     | 0         | 54          | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| permdata   | cp_79                              | 3     | 0         | 0           | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pem        | dashboard_settings                 | 3     | 0         | 173990      | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| permdata   | streaming_replication_db_conflicts | 3     | 0         | 0           | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| permdata   | efm_cluster_info                   | 3     | 0         | 757         | 0               | 0             | 0            | 0            | 0                | 0          | 0         |

## Top 20 Indexes Activity

| Schema | Table Name        | Index Name                 | Scans   | Rows Read | Rows Fetched | Blocks Read | Blocks Hit |
|--------|-------------------|----------------------------|---------|-----------|--------------|-------------|------------|
| pem    | chart             | pem_chart_pk               | 383     | 383       | 205          | 45          | 777        |
| pem    | config            | config_pkey                | 58      | 58        | 58           | 36          | 158        |
| pem    | server            | server_pkey                | 5951621 | 5843186   | 5843186      | 45          | 5912350    |
| pem    | server_group      | server_group_pkey          | 0       | 0         | 0            | 28          | 119        |
| pem    | server_group      | uniqueconstraint           | 481     | 482       | 482          | 41          | 583        |
| pem    | user_server_group | user_server_group_pk       | 432     | 0         | 0            | 21          | 1642       |
| pem    | server_options    | server_option_pkey         | 0       | 0         | 0            | 39          | 295        |
| pem    | server_options    | server_option_pem_user_key | 2243    | 2243      | 2237         | 50          | 2524       |
| pem    | server_auth       | server_auth_pkey           | 0       | 0         | 0            | 31          | 105        |

|     |                      |                               |          |          |          |    |          |
|-----|----------------------|-------------------------------|----------|----------|----------|----|----------|
| pem | server_auth          | server_auth_pem_user_key      | 448      | 448      | 448      | 38 | 542      |
| pem | database_option      | database_option_pkey          | 0        | 0        | 0        | 3  | 773      |
| pem | database_option      | database_option_server_id_key | 763      | 0        | 0        | 3  | 2299     |
| pem | agent                | agent_pkey                    | 47058865 | 46533522 | 46533522 | 47 | 47059162 |
| pem | agent_server_binding | agent_server_binding_pkey     | 25262493 | 22384663 | 18590220 | 48 | 25262808 |
| pem | agent_options        | agent_option_pkey             | 52070    | 16528    | 13761    | 38 | 490680   |
| pem | agent_options        | agent_option_pem_user_key     | 1        | 2        | 0        | 31 | 436451   |
| pem | server_version       | server_version_pkey           | 575      | 855      | 727      | 17 | 603      |
| pem | probe_target_type    | probe_target_type_pkey        | 1835627  | 1835627  | 1835627  | 41 | 1835812  |
| pem | agent_heartbeat      | agent_heartbeat_agent_id_key  | 111      | 110      | 110      | 27 | 2666     |
| pem | server_heartbeat     | server_heartbeat_uniq         | 53       | 51       | 51       | 33 | 2097     |
| pem | probe                | probe_pkey                    | 1225110  | 1246866  | 1246794  | 43 | 1245002  |
| pem | probe_server_version | probe_server_version_pkey     | 0        | 0        | 0        | 60 | 992      |
| pem | probe_server_version | probe_server_version_unique   | 33752622 | 82437234 | 82123290 | 92 | 69074459 |
| pem | probe_column         | probe_parameter_pkey          | 71       | 71       | 71       | 52 | 741      |
| pem | probe_column         | probe_parameter_unique        | 512246   | 2214060  | 148323   | 81 | 1024971  |

Use parameters on the [PEM Server Configurations](#) dialog to specify the auto-refresh rate for the dashboard. To access the [Server Configuration](#) dialog, select [Server Configuration...](#) from the PEM web interface [Management](#) menu.

The I/O Analysis dashboard header displays the date and time that the server started, the date and time that the I/O Analysis dashboard was last updated, and the number of alerts currently triggered for the specified database (and any monitored object that resides within that database).

The graphs in the [I/O Overview](#) provide information about the week's activity for the specified database:

- The [Database I/O](#) graph displays the number of blocks read to and written from disk and memory buffers for the specified database over the course of the previous week.

The vertical key on the left side of the graph charts the block count.

- The [Row Activity](#) graph displays tuple activity for tables residing within the database over the last week.

The vertical key on the left side of the graph charts the row count.

- The [Checkpoints](#) graph displays the number of timed and untimed (requested) checkpoints written for the database over the last week.

The vertical key on the left side of the graph displays the checkpoint count.

A checkpoint is a point in the transaction logging sequence at which all data files have been updated to reflect the information in the log, and data files are flushed to disk. Checkpoints can be automatically generated, or forced by use of the CHECKPOINT command. A timed checkpoint occurs when the checkpoints\_timeout parameter time limit is met. An untimed (requested) checkpoint occurs when the checkpoint\_segments parameter is met, or when a superuser issues the CHECKPOINT command. Frequent checkpointing can impose extra load on the server, but can reduce recovery time in the event of a crash or hardware failure.

The [Hot Tables/Indexes](#) section of the I/O Analysis dashboard provides an overview of the 5 most scanned tables and indexes that reside within the database.

- The [Hot Tables](#) bar graph represents the comparative usage of the 5 most scanned tables that reside in the database; a vertical key displays the number of table scans.
- The [Hot Indexes](#) bar graph represents the comparative usage of the 5 most scanned indexes that reside in the database; a vertical key displays the number of index scans.

The [Object I/O Details](#) section of the I/O Analysis dashboard provides tables that display the table and index activity for the selected database.

The [Tables Activity](#) table provides a detailed analysis of the activity for the 20 most active tables that reside within the database. Click a column heading to sort the table by the values within the column; click again to reverse the sort order.



| Object I/O Details |                                    |       |           |             |                 |               |              |              |                  |            |           |
|--------------------|------------------------------------|-------|-----------|-------------|-----------------|---------------|--------------|--------------|------------------|------------|-----------|
| Objects Activity   |                                    |       |           |             |                 |               |              |              |                  |            |           |
| Schema             | Table Name                         | Scans | Rows Read | Index Scans | Index Rows Read | Rows Inserted | Rows Updated | Rows Deleted | Hot Rows Updated | Total Rows | Dead Rows |
| pemdata            | cp_81                              | 1     | 0         | 2561        | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pemhistory         | cp_81                              | 2     | 0         | 0           | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pem                | bart_log                           | 2     | 0         |             |                 | 0             | 0            | 0            | 0                | 0          | 0         |
| pem                | bart_host_public_key               | 2     | 0         |             |                 | 0             | 0            | 0            | 0                | 0          | 0         |
| pem                | data_chart                         | 3     | 6         | 267         | 250             | 10            | 0            | 1            | 0                | 3          | 7         |
| pem                | bart_config                        | 3     | 0         | 0           | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pemdata            | cp_80                              | 3     | 0         | 0           | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pemdata            | function_statistics                | 3     | 0         | 5600        | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pem                | bart_server_config                 | 3     | 0         | 0           | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pem                | bart_default_config                | 3     | 56        | 0           | 0               | 84            | 0            | 0            | 0                | 28         | 0         |
| pemdata            | number_of_prepared_transactions    | 3     | 10        | 57076       | 56617           | 16            | 0            | 0            | 0                | 6          | 10        |
| pem                | tbl_chart                          | 3     | 26        | 108500      | 284             | 40            | 0            | 1            | 0                | 13         | 27        |
| pemdata            | oc_tablespace                      | 3     | 20        | 8937        | 17862           | 32            | 0            | 0            | 0                | 12         | 20        |
| pem                | ark_server_option                  | 3     | 0         | 0           | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pem                | server_version                     | 3     | 50        | 575         | 855             | 75            | 0            | 0            | 0                | 25         | 50        |
| pem                | probe_config_view                  | 3     | 0         | 54          | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pemdata            | cp_79                              | 3     | 0         | 0           | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pem                | dashboard_settings                 | 3     | 0         | 174298      | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pemdata            | streaming_replication_db_conflicts | 3     | 0         | 0           | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pemdata            | efm_cluster_info                   | 3     | 0         | 803         | 0               | 0             | 0            | 0            | 0                | 0          | 0         |

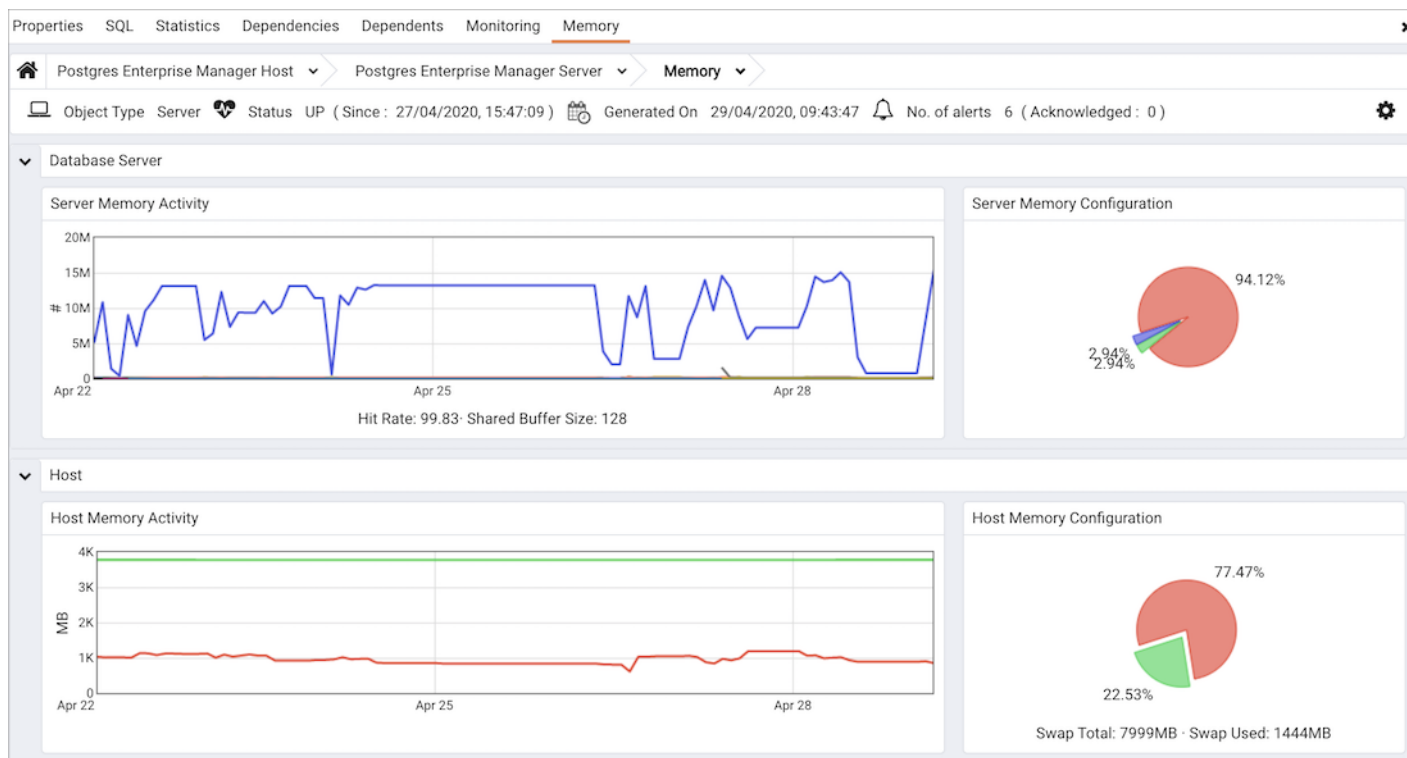
- The **Schema** column identifies the schema in which the table resides.
- The **Table Name** column identifies the name of the table.
- The **Scans** column displays the number of scans performed on the table.
- The **Rows Read** column displays the number of rows read from the specified table.
- The **Index Scans** column displays the number of index scans performed on the specified table.
- The **Index Rows Read** column displays the number of rows read during index scans on the specified table.
- The **Rows Inserted** column displays the number of rows inserted into the specified table.
- The **Rows Updated** column displays the number of rows updated in the specified table.
- The **Rows Deleted** column displays the number of rows deleted from the specified table.
- The **Hot Rows Updated** column displays the number of hot row updates for the table; when a hot row update occurs, the new row occupies the same page as the previous row.
- The **Total Rows** column displays the number of total rows in the table.
- The **Dead Rows** column displays the number of rows that have been deleted, but have not been reclaimed via a VACUUM command or the AUTOVACUUM process.

The **Indexes Activity** table provides a detailed analysis of the activity for the 20 most active indexes. Click a column heading to sort the table by the values within the column; click again to reverse the sort order.

- The **Schema** column identifies the schema in which the index resides.
- The **Table Name** column identifies the name of the table on which the index is defined.
- The **Index Name** column displays the name of the index.
- The **Scans** column displays the number of index scans performed on the specified table.
- The **Rows Read** column displays the number of tuples read during index scans on the specified table.
- The **Rows Fetched** column displays the number of tuples fetched by index scans.
- The **Blocks Read** column displays the number of index blocks read.
- The **Blocks Hit** column displays the number of index blocks hit.

### 34.4.1.7 The Memory Analysis Dashboard

The **Memory Analysis** dashboard provides an overview of the memory usage for the selected server and server host for the previous week:



Use parameters on the [PEM Server Configurations](#) dialog to specify the auto-refresh rate for the dashboard. To access the [Server Configuration](#) dialog, select [Server Configuration...](#) from the PEM web interface [Management](#) menu.

The Memory Analysis dashboard header displays the date and time that the server was started, the date and time that the dashboard was last updated and the number of current alerts for objects monitored by the PEM server.

The [Database Server](#) section displays memory usage trends for the selected server.

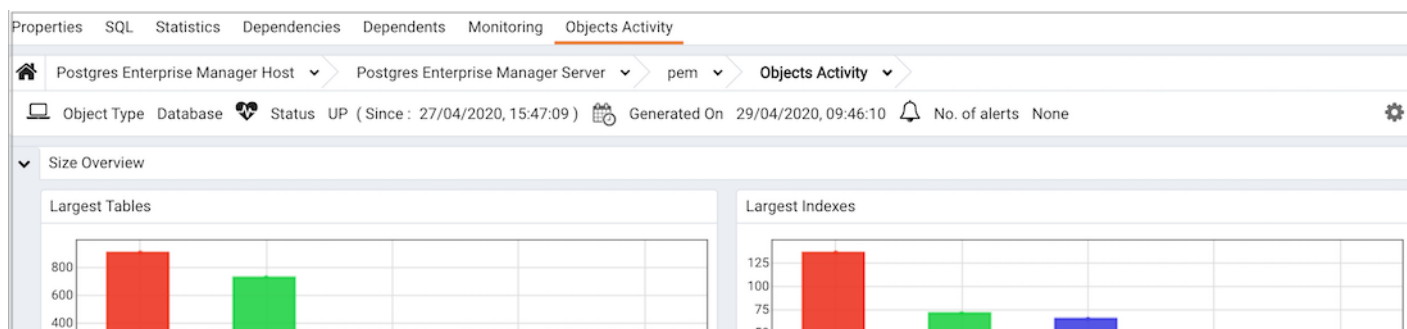
- The [Server Memory Activity](#) graph displays the previous week's activity on the server; the [Legend](#) at the bottom of the graph provide a key to the colors used to chart information for each database. A vertical key on the left side of the graph indicates the actual block count for each value.
- The [Server Memory Configuration](#) pie chart displays the current memory usage (in megabytes).

The [Host](#) section displays the free and used memory on the host system:

- The [Host Memory Activity](#) chart plots the free and used memory on the host system over the last week.
- Sections of the [Host Memory Configuration](#) pie chart represent the free and available memory on the host system when the last probe executed.

### 34.4.1.8 The Objects Activity Analysis Dashboard

The Objects Activity Analysis dashboard provides an overview of the size and activity of the objects that reside within the selected database.





Objects Activity

| Schema     | Table Name                         | Scans | Rows Read | Index Scans | Index Rows Read | Rows Inserted | Rows Updated | Rows Deleted | Hot Rows Updated | Total Rows | Dead Rows |
|------------|------------------------------------|-------|-----------|-------------|-----------------|---------------|--------------|--------------|------------------|------------|-----------|
| pemdata    | cp_81                              | 1     | 0         | 2561        | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pemhistory | cp_81                              | 2     | 0         | 0           | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pem        | bart_log                           | 2     | 0         |             |                 | 0             | 0            | 0            | 0                | 0          | 0         |
| pem        | bart_host_public_key               | 2     | 0         |             |                 | 0             | 0            | 0            | 0                | 0          | 0         |
| pem        | data_chart                         | 3     | 6         | 267         | 250             | 10            | 0            | 1            | 0                | 3          | 7         |
| pem        | bart_config                        | 3     | 0         | 0           | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pemdata    | cp_80                              | 3     | 0         | 0           | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pemdata    | function_statistics                | 3     | 0         | 5600        | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pem        | bart_server_config                 | 3     | 0         | 0           | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pem        | bart_default_config                | 3     | 56        | 0           | 0               | 84            | 0            | 0            | 0                | 28         | 0         |
| pemdata    | number_of_prepared_transactions    | 3     | 10        | 57076       | 56617           | 16            | 0            | 0            | 0                | 6          | 10        |
| pem        | tbl_chart                          | 3     | 26        | 108500      | 284             | 40            | 0            | 1            | 0                | 13         | 27        |
| pemdata    | oc_tablespace                      | 3     | 20        | 8937        | 17862           | 32            | 0            | 0            | 0                | 12         | 20        |
| pem        | ark_server_option                  | 3     | 0         | 0           | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pem        | server_version                     | 3     | 50        | 575         | 855             | 75            | 0            | 0            | 0                | 25         | 50        |
| pem        | probe_config_view                  | 3     | 0         | 54          | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pemdata    | cp_79                              | 3     | 0         | 0           | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pem        | dashboard_settings                 | 3     | 0         | 174298      | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pemdata    | streaming_replication_db_conflicts | 3     | 0         | 0           | 0               | 0             | 0            | 0            | 0                | 0          | 0         |
| pemdata    | efm_cluster_info                   | 3     | 0         | 803         | 0               | 0             | 0            | 0            | 0                | 0          | 0         |

Object Storage

| Schema     | Object                             | Object Type | Table Size (MB) | Index Size (MB) | Total(MB) |
|------------|------------------------------------|-------------|-----------------|-----------------|-----------|
| pemhistory | session_info_keyidx                | Index       |                 | 11              | 11        |
| pemdata    | oc_function_pkey                   | Index       |                 | 10              | 10        |
| pemdata    | server_logs_command_tag_idx        | Index       |                 | 36              | 36        |
| pemdata    | server_logs_error_severity_idx     | Index       |                 | 36              | 36        |
| pemdata    | server_logs_logtime_by_minutes_idx | Index       |                 | 38              | 38        |
| pemdata    | server_logs_log_time_idx           | Index       |                 | 38              | 38        |
| pemdata    | server_logs_pkey                   | Index       |                 | 38              | 38        |
| pemdata    | server_logs_server_id_idx          | Index       |                 | 36              | 36        |
| pemhistory | index_size_keyidx                  | Index       |                 | 1               | 1         |
| pemhistory | index_statistics_keyidx            | Index       |                 | 65              | 65        |
| pemhistory | index_statistics_timeidx           | Index       |                 | 26              | 26        |
| pemhistory | lock_info_keyidx                   | Index       |                 | 26              | 26        |
| pemhistory | lock_info_timeidx                  | Index       |                 | 14              | 14        |
| pemhistory | oc_function_keyidx                 | Index       |                 | 21              | 21        |
| pemhistory | oc_function_timeidx                | Index       |                 | 4               | 4         |
| pemhistory | session_info_timeidx               | Index       |                 | 8               | 8         |
| pemhistory | session_waits_keyidx               | Index       |                 | 1               | 1         |
| pemhistory | table_frozenxid_keyidx             | Index       |                 | 8               | 8         |
| pemhistory | table_frozenxid_timeidx            | Index       |                 | 3               | 3         |
| pemhistory | table_statistics_keyidx            | Index       |                 | 136             | 136       |
| pemhistory | table_statistics_timeidx           | Index       |                 | 71              | 71        |
| pemhistory | table_statistics                   | Table       | 904             | 208             | 1113      |
| pemdata    | server_logs                        | Table       | 727             | 221             | 949       |
| pemhistory | index_statistics                   | Table       | 151             | 92              | 243       |
| pemhistory | session_info                       | Table       | 118             | 20              | 139       |
| pemhistory | lock_info                          | Table       | 49              | 40              | 90        |
| pem        | probe_log                          | Table       | 47              | 0               | 47        |
| pemhistory | oc_function                        | Table       | 22              | 25              | 47        |
| pemdata    | oc_function                        | Table       | 10              | 10              | 21        |
| pemhistory | table_frozenxid                    | Table       | 10              | 11              | 22        |

|            |                              |       |   |   |    |
|------------|------------------------------|-------|---|---|----|
| pemhistory | oc_views                     | Table | 8 | 1 | 10 |
| pemdata    | oc_views                     | Table | 4 | 0 | 4  |
| pemhistory | session_waits                | Table | 3 | 2 | 6  |
| pemhistory | background_writer_statistics | Table | 3 | 1 | 4  |
| pemhistory | database_statistics          | Table | 3 | 1 | 5  |
| pemhistory | server_info                  | Table | 2 | 0 | 3  |
| pemhistory | log_configuration            | Table | 2 | 0 | 3  |
| pemhistory | index_size                   | Table | 1 | 1 | 2  |
| pemhistory | cpu_usage                    | Table | 1 | 1 | 3  |
| pemhistory | network_statistics           | Table | 1 | 0 | 2  |
| pemhistory | oc_index                     | Table | 1 | 1 | 2  |
| pemhistory | memory_usage                 | Table | 1 | 0 | 1  |
| pemdata    | table_statistics             | Table | 1 | 0 | 2  |
| pemhistory | table_bloat                  | Table | 1 | 0 | 2  |
| pem        | jobsteplog                   | Table | 0 | 0 | 1  |
| pemhistory | table_size                   | Table | 0 | 0 | 1  |
| pemdata    | index_size                   | Table | 0 | 0 | 1  |
| pemdata    | index_statistics             | Table | 0 | 0 | 1  |
| pemhistory | user_info                    | Table | 0 | 0 | 1  |
| pemhistory | oc_table                     | Table | 0 | 0 | 1  |

Use parameters on the [PEM Server Configurations](#) dialog to specify the auto-refresh rate for the dashboard. To access the [Server Configuration](#) dialog, select [Server Configuration...](#) from the PEM web interface [Management](#) menu.

The Objects Activity Analysis dashboard header displays the date and time that the server started, the date and time that the Object Activity Analysis dashboard was last updated, and the number of alerts currently triggered for the specified database (and monitored objects that reside within that database).

The bar graphs in the [Size Overview](#) section plot the comparative sizes of the 5 largest tables and indexes that reside within the selected database:

- The [Top 5 Largest Tables](#) bar graph represents the comparative sizes of the 5 largest tables that reside in the database; a vertical key displays the table size in megabytes.
- The [Top 5 Largest Indexes](#) bar graph represents the comparative sizes of the 5 largest indexes that reside in the database; a vertical key displays the index size in megabytes.

The [Objects Activity](#) table provides a detailed analysis of the activity for each table that resides within the database. Click a column heading to sort the table by the values within the column; click again to reverse the sort order.

- The [Schema](#) column identifies the schema in which the specified table resides.
- The [Table Name](#) column identifies the name of the table.
- The [Scans](#) column displays the number of scans performed on the table.
- The [Rows Read](#) column displays the number of rows read from the specified table.
- The [Index Scans](#) column displays the number of index scans performed on the specified table.
- The [Index Rows Read](#) column displays the number of rows read during index scans on the specified table.
- The [Rows Inserted](#) column displays the number of rows inserted into the specified table.
- The [Rows Updated](#) column displays the number of rows updated in the specified table.
- The [Rows Deleted](#) column displays the number of rows deleted from the specified table.
- The [Hot Rows Updated](#) column displays the number of hot row updates into the table; when a hot row update occurs, the new row occupies the same page as the previous row.
- The [Total Rows](#) column displays the number of total rows in the table.
- The [Dead Rows](#) column displays the number of rows that have been deleted, but have not been reclaimed via a VACUUM command or the AUTOVACUUM process.

The [Objects Storage](#) table displays the schema objects that reside in the selected database. Click a column heading to sort the table data by the values within that column; click again to reverse the sort order.

| Object Storage |                                    |             |                 |                 |           |
|----------------|------------------------------------|-------------|-----------------|-----------------|-----------|
| Schema         | Object                             | Object Type | Table Size (MB) | Index Size (MB) | Total(MB) |
| pemhistory     | session_info_keyidx                | Index       |                 | 11              | 11        |
| pemdata        | oc_function_pkey                   | Index       |                 | 10              | 10        |
| pemdata        | server_logs_command_tag_idx        | Index       |                 | 36              | 36        |
| pemdata        | server_logs_error_severity_idx     | Index       |                 | 36              | 36        |
| pemdata        | server_logs_logtime_by_minutes_idx | Index       |                 | 38              | 38        |
| pemdata        | server_logs_log_time_idx           | Index       |                 | 38              | 38        |
| pemdata        | server_logs_pkey                   | Index       |                 | 38              | 38        |
| pemdata        | server_logs_server_id_idx          | Index       |                 | 36              | 36        |
| pemhistory     | index_size_keyidx                  | Index       |                 | 1               | 1         |
| pemhistory     | index_statistics_keyidx            | Index       |                 | 65              | 65        |
| pemhistory     | index_statistics_timeidx           | Index       |                 | 26              | 26        |
| pemhistory     | lock_info_keyidx                   | Index       |                 | 26              | 26        |
| pemhistory     | lock_info_timeidx                  | Index       |                 | 14              | 14        |
| pemhistory     | oc_function_keyidx                 | Index       |                 | 21              | 21        |
| pemhistory     | oc_function_timeidx                | Index       |                 | 4               | 4         |
| pemhistory     | session_info_timeidx               | Index       |                 | 8               | 8         |
| pemhistory     | session_waits_keyidx               | Index       |                 | 1               | 1         |
| pemhistory     | table_frozenxid_keyidx             | Index       |                 | 8               | 8         |
| pemhistory     | table_frozenxid_timeidx            | Index       |                 | 3               | 3         |
| pemhistory     | table_statistics_keyidx            | Index       |                 | 136             | 136       |
| pemhistory     | table_statistics_timeidx           | Index       |                 | 71              | 71        |
| pemhistory     | table_statistics                   | Table       | 904             | 208             | 1113      |
| pemdata        | server_logs                        | Table       | 727             | 221             | 949       |
| pemhistory     | index_statistics                   | Table       | 151             | 92              | 243       |
| pemhistory     | session_info                       | Table       | 118             | 20              | 139       |
| pemhistory     | lock_info                          | Table       | 49              | 40              | 90        |
| pem            | probe_log                          | Table       | 47              | 0               | 47        |
| pemhistory     | oc_function                        | Table       | 22              | 25              | 47        |
| pemdata        | oc_function                        | Table       | 10              | 10              | 21        |
| pemhistory     | table_frozenxid                    | Table       | 10              | 11              | 22        |
| pemhistory     | oc_views                           | Table       | 8               | 1               | 10        |
| pemdata        | oc_views                           | Table       | 4               | 0               | 4         |
| pemhistory     | session_waits                      | Table       | 3               | 2               | 6         |
| pemhistory     | background_writer_statistics       | Table       | 3               | 1               | 4         |
| pemhistory     | database_statistics                | Table       | 3               | 1               | 5         |
| pemhistory     | server_info                        | Table       | 2               | 0               | 3         |
| pemhistory     | log_configuration                  | Table       | 2               | 0               | 3         |
| pemhistory     | index_size                         | Table       | 1               | 1               | 2         |
| pemhistory     | cpu_usage                          | Table       | 1               | 1               | 3         |
| pemhistory     | network_statistics                 | Table       | 1               | 0               | 2         |
| pemhistory     | oc_index                           | Table       | 1               | 1               | 2         |
| pemhistory     | memory_usage                       | Table       | 1               | 0               | 1         |
| pemdata        | table_statistics                   | Table       | 1               | 0               | 2         |
| pemhistory     | table_bloat                        | Table       | 1               | 0               | 2         |
| pem            | jobsteplog                         | Table       | 0               | 0               | 1         |
| pemhistory     | table_size                         | Table       | 0               | 0               | 1         |
| pemdata        | index_size                         | Table       | 0               | 0               | 1         |
| pemdata        | index_statistics                   | Table       | 0               | 0               | 1         |
| pemhistory     | user_info                          | Table       | 0               | 0               | 1         |
| pemhistory     | oc_table                           | Table       | 0               | 0               | 1         |

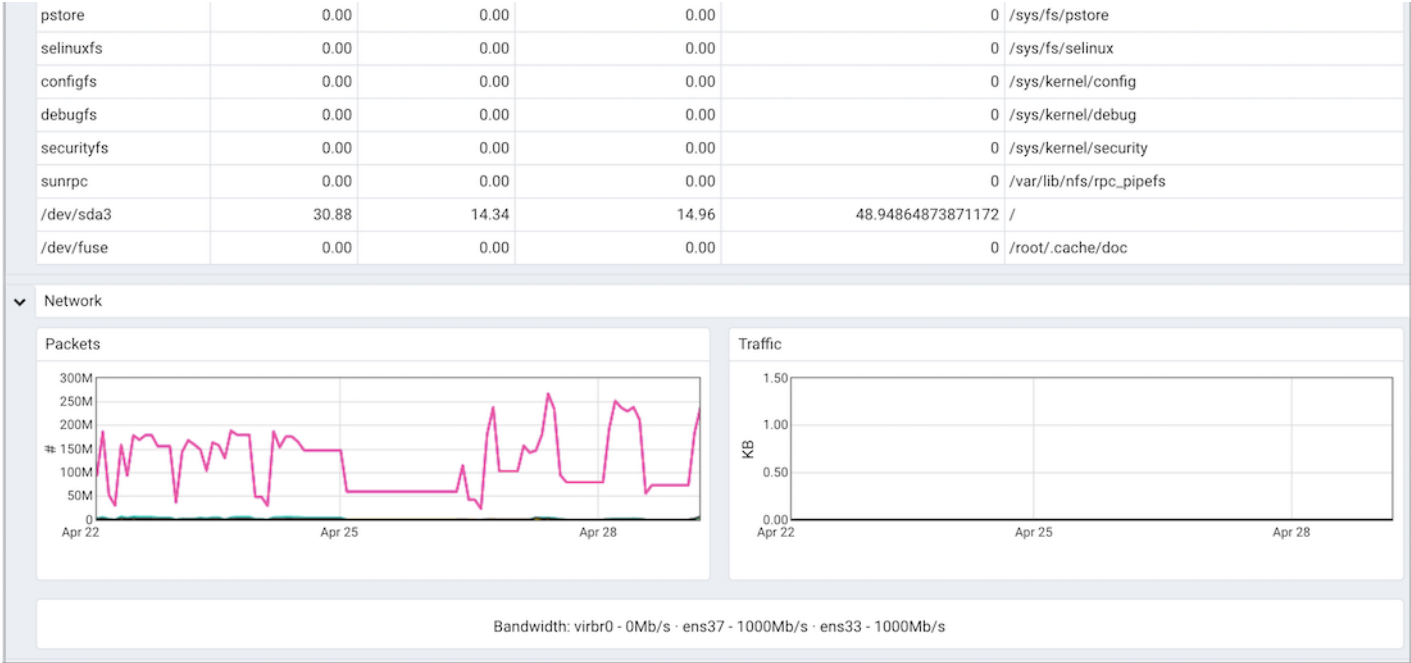
- The **Schema** column identifies the schema in which the object resides.
- The **Object** column identifies the name of the schema object.
- The **Object Type** column identifies the type of schema object (Table or Index).
- The **Table Size** column lists the size of the table in megabytes (if applicable).
- The **Index Size** column lists the size of the index (or associated index) in megabytes (if applicable).

- The **Total (MB)** column lists the cumulative size (in megabytes) of the specified table and/or indexes and associated TOAST tables.

### 34.4.1.9 The Operating System Analysis Dashboard

The **Operating System Analysis** dashboard provides a graphical analysis of the resource usage on the system hosting the selected agent.





Use parameters on the [PEM Server Configurations](#) dialog to specify the auto-refresh rate for the dashboard. To access the [Server Configuration](#) dialog, select [Server Configuration...](#) from the PEM web interface [Management](#) menu.

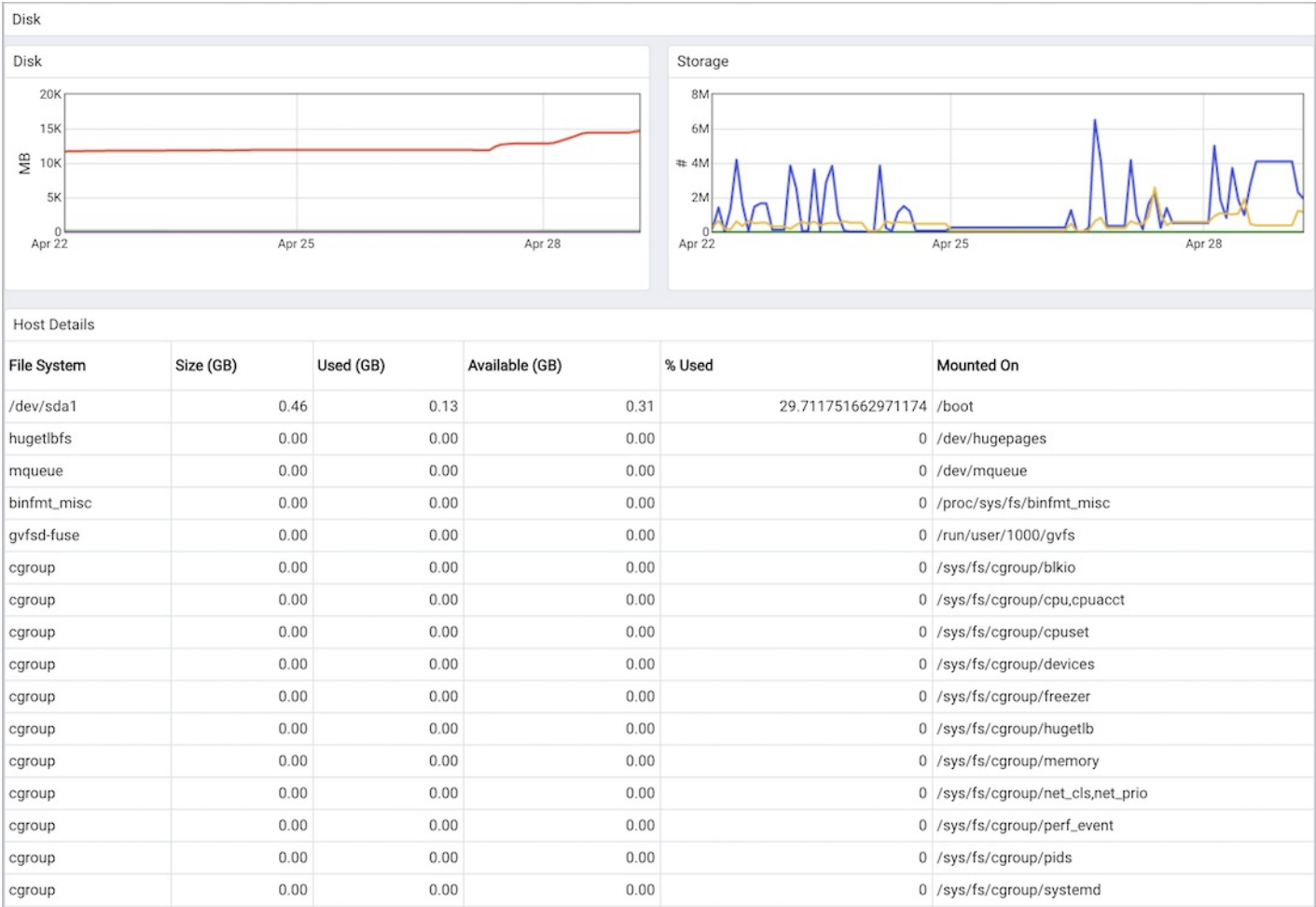
The [Operating System Analysis](#) dashboard header displays the date and time that the server was last booted, the date and time that the display was last updated, and the number of triggered alerts on the system.

The Operating System Analysis dashboard provides an overview of system resources. Within the [OS Overview](#) section:

- The [CPU](#) graph represents the percentage of the CPU used at a given point in time. The vertical key on the left side of the graph indicates the percentage.
- Segments of the [Storage](#) pie chart represent the free and used storage on the host.
- The [Memory](#) graph displays the memory usage on the PEM server.
- The [Process](#) graph plots the number of processes on the system. A vertical key on the left side of the graph displays the process count.

The [Disk](#) section of the [Operating System Analysis](#) dashboard displays charts and information about operating system disk usage.





- The **Disk** graph displays the amount of disk space used. The vertical key on the left side of the chart displays the amount of disk space used (in Megabytes). Each horizontal line on the graph represents a different mounted file system; a file system key is provided in the **Legend**.
- The **I/O** graph displays the blocks read from and written to disk. A vertical key on the left side of the graph provides a block count.
- The **Host File System Details** table provides information about the host file system:
  - The **File System** column displays the name of the file system.
  - The **Size (GB)** column displays the size of the file system in Gigabytes.
  - The **Used (GB)** column displays the amount of the file system that is currently storing information.
  - The **Available (GB)** column displays the amount of space still available on the file system.
  - The **% Used** column displays the percentage of the total storage space in use.
  - The **Mounted On** column displays the directory or drove on which the file system is mounted.

Graphs in the **Network** section of the **Operating System Analysis** dashboard plot the network and packet traffic:





- The **Packets** graph displays the number of packets sent and received across the network. The **Legend** provides a key to the color charted for each network interface. The vertical key on the left side of the graph indicates the packet count.
- The **Traffic** graph displays the amount of data transferred across the network. The **Legend** provides a key to the color charted for each network interface. The vertical key on the left side of the graph displays the traffic, in KB.

Please note: The network bandwidth may not display if the monitored server is a Linux platform that resides in a virtual machine. This is expected behavior.

### 34.4.1.10 The Probe Log Analysis Dashboard

The Probe Log Analysis dashboard displays error messages from the PEM agent.

| Properties SQL Statistics Dependencies Dependents Monitoring <b>Probe Log</b>  |                      |                          |                                    |   |  |
|--|----------------------|--------------------------|------------------------------------|---|--|
| Postgres Enterprise Manager Host ▾ Probe Log ▾   |                      |                          |                                    |   |  |
| Object Type Host Agent Status UP ( Since : 15/04/2020, 19:35:06 ) Generated On 29/04/2020, 10:07:25 No. of alerts None |                      |                          |                                    |   |  |
| Probe Logs   |                      |                          |                                    |   |  |
| id   | Timestamp            | Probe Name               | Server                             | Error Message                                       |  |
| 116854   | 29/04/2020, 10:05:59 | Object Catalog: View     | Postgres Enterprise Manager Server | ERROR: canceling statement due to statement timeout |  |
| 116849   | 29/04/2020, 10:05:38 | Object Catalog: View     | Postgres Enterprise Manager Server | ERROR: canceling statement due to statement timeout |  |
| 116765   | 29/04/2020, 10:03:39 | Object Catalog: Function | Postgres Enterprise Manager Server | ERROR: canceling statement due to statement timeout |  |
| 116764   | 29/04/2020, 10:03:37 | Object Catalog: View     | Postgres Enterprise Manager Server | ERROR: canceling statement due to statement timeout |  |
| 116757   | 29/04/2020, 10:00:40 | Object Catalog: View     | Postgres Enterprise Manager Server | ERROR: canceling statement due to statement timeout |  |
| 116677   | 29/04/2020, 09:56:04 | Object Catalog: View     | Postgres Enterprise Manager Server | ERROR: canceling statement due to statement timeout |  |
| 116669   | 29/04/2020, 09:55:34 | Object Catalog: View     | Postgres Enterprise Manager Server | ERROR: canceling statement due to statement timeout |  |
| 116637   | 29/04/2020, 09:54:03 | Table Bloat              | Postgres Enterprise Manager Server | ERROR: canceling statement due to statement timeout |  |
| 116585   | 29/04/2020, 09:51:01 | Object Catalog: Function | Postgres Enterprise Manager Server | ERROR: canceling statement due to statement timeout |  |
| 116584   | 29/04/2020, 09:51:01 | Object Catalog: View     | Postgres Enterprise Manager Server | ERROR: canceling statement due to statement timeout |  |
| 116578   | 29/04/2020, 09:50:40 | Object Catalog: View     | Postgres Enterprise Manager Server | ERROR: canceling statement due to statement timeout |  |
| 116503   | 29/04/2020, 09:46:08 | Table Statistics         | Postgres Enterprise Manager Server | ERROR: canceling statement due to statement timeout |  |
| 116497   | 29/04/2020, 09:45:57 | Object Catalog: Function | Postgres Enterprise Manager Server | ERROR: canceling statement due to statement timeout |  |
| 116496   | 29/04/2020,          | Object Catalog: View     | Postgres Enterprise                | ERROR: canceling statement due to statement timeout |  |

The header information includes the date and time that the server was first started, the date and time that the page was last updated, and the current number of triggered alerts.

Use parameters on the **PEM Server Configurations** dialog to specify the auto-refresh rate for the dashboard. To access the **Server Configuration** dialog, select **Server Configuration...** from the PEM web interface **Management** menu.

The **Probe Log** table displays error messages returned by the PEM Agent. Entries in the Probe Log table may reflect incorrect agent binding information or authentication errors between the PEM agent and the server.

- The **Id** column displays a unique identifier for each entry in the table.
- The **Timestamp** column displays the date and time that the log entry was made.
- The **Probe Name** column displays the name of the probe that recorded the log entry.
- The **Server Name** column displays the name of the server on which the error occurred.
- The **Error Message** column displays the error message returned by the probe.

### 34.4.1.11 The Server Analysis Dashboard

The Server Analysis dashboard provides a graphical analysis of a monitored server's usage statistics.



The Server Analysis dashboard header displays the date and time that the server was started, the date and time that the display was last updated, and the number of current alerts for items monitored by the PEM server.

Use parameters on the [PEM Server Configurations](#) dialog to specify the auto-refresh rate for the dashboard. To access the [Server Configuration](#) dialog, select [Server Configuration...](#) from the PEM web interface [Management](#) menu.

Graphs within the [Storage](#) section of the dashboard provide an analysis of the space consumed by databases and tablespaces on the server:

- The [Database Size](#) graph displays the size (in Megabytes) of the 5 largest databases that reside on the PEM server. The [Legend](#) at the bottom of the graph associates each database name with a color in the graph.
- The [Tablespace Size](#) graph displays the size (in Megabytes) of the 5 largest tablespaces that reside on the PEM server. The [Legend](#) at the bottom of the graph associates each tablespace name with a color in the graph.

The [Memory](#) section of the dashboard provides an overview of the efficiency of the buffer cache over the previous week, and an analysis of the current swap memory usage:

- The [Shared Buffers](#) chart compares the number of data blocks found in the shared memory cache with the number of blocks read from disk. A high hit-to-miss ratio indicates an efficiently configured memory cache.
- The [Host Memory](#) pie chart displays the current swap memory usage.

The [Users](#) section of the [Server Analysis](#) dashboard provides an overview of the user activity on the server:

- The [User Activity](#) chart displays connection statistics gathered over the last week. The [Legend](#) at the bottom of the chart provides a key to the data displayed.
- The [Connection Overview](#) pie chart compares the currently active connections to the currently idle connections.

The [I/O](#) section of the [Server Analysis](#) dashboard provides an overview of the transactions processed by the server over the last week:



- The [Disk](#) chart displays the number of 8KB blocks read from disk, and the number of 8KB blocks written to disk over the last week.
- The [Row Activity](#) chart plots row activity on tables stored on the server over the past week. The [Legend](#) at the bottom of the chart provides a key to the data displayed.
- The [Commits/Rollbacks](#) chart displays the number of transactions committed and rolled back on the selected server within the last week. A vertical count on the left side of the graph indicates the aborted transaction count, while the [Legend](#) at the bottom of the chart provides a key to the commits and rollbacks charted.

The [Database Analysis](#) table displays a list of the monitored databases that reside on the server, and the statistics gathered for each database over the last week. Click a column heading to sort the table by the data displayed in the column; click again to reverse the sort order.

| Databases            |             |              |                |            |             |                |                 |                 |                |                |
|----------------------|-------------|--------------|----------------|------------|-------------|----------------|-----------------|-----------------|----------------|----------------|
| Databases Statistics |             |              |                |            |             |                |                 |                 |                |                |
| Database             | Connections | TX Committed | TX Rolled Back | Blocks Hit | Blocks Read | Tuples Fetched | Tuples Returned | Tuples Inserted | Tuples Updated | Tuples Deleted |
| postgres             | 3           | 1148192      | 22             | 139399784  | 5982        | 61900078       | 61900078        | 67              | 15             | 0              |
| pem                  | 16          | 13239743     | 5840           | 7607117142 | 12978313    | 1475568227     | 1475568227      | 7506885         | 7929549        | 460295         |
| hr                   | 0           | 67207        | 1621           | 13155426   | 1632        | 5505911        | 5505911         | 0               | 0              | 0              |
| edbstore_temp        | 0           | 69973        | 0              | 14834311   | 1453        | 6529192        | 6529192         | 32              | 1              | 0              |
| db01                 | 0           | 115198       | 29             | 15536425   | 2361        | 6817777        | 6817777         | 374             | 28             | 12             |
| testdb               | 0           | 160130       | 0              | 11070985   | 6764        | 2573029        | 2573029         | 238866          | 416394         | 0              |

- The **Database** column displays the database name.
- The **Connections** column displays the number of current connections to the database.
- The **TX Committed** column displays the number of transactions committed to the database within the last week.
- The **TX Rolled Back** column displays the number of transactions rolled back within the last week.
- The **Blocks Hit** column displays the number of blocks hit in the cache (in megabytes) within the last week.
- The **Blocks Read** column displays the number of blocks read from memory (in megabytes) within the last week.
- The **Tuples Fetched** column displays the number of tuples fetched within the last week.
- The **Tuples Returned** column displays the number of tuples returned within the last week.
- The **Tuples Inserted** column displays the number of tuples inserted into the database within the last week.
- The **Tuples Updated** column displays the number of tuples updated in the database within the last week.
- The **Tuples Deleted** column displays the number of tuples deleted from the database within the last week.

### 34.4.1.12 The Server Log Analysis Dashboard

The **Server Log Analysis** dashboard displays the log files for the selected server. To view the **Server Log Analysis** dashboard, right-click on the name of a monitored server in the PEM client tree control, and navigate through the **Dashboards** menu, selecting **Server Log Analysis**.

Properties SQL Statistics Dependencies Dependents Monitoring **Server Log**

Postgres Enterprise Manager Host Postgres Enterprise Manager Server **Server Log**

Object Type Server Status UP ( Since : 27/04/2020, 15:47:09 ) Generated On 29/04/2020, 10:20:00 No. of alerts 6 ( Acknowledged : 0 )

Server Logs

| id      | Timestamp            | User Name | Database Name | Process ID | Session ID    | Transaction ID | Connection From | Command | Message   |
|---------|----------------------|-----------|---------------|------------|---------------|----------------|-----------------|---------|---|
| 1870601 | 29/04/2020, 10:15:06 | agent1    | pem           | 64373      | 5ea8fbc6.fb75 | 5002111        | 127.0.0.1:55512 | COPY    | duration: 0.187 ms statement: BEGIN;COPY pemdata.server_logs(server_id, log_time, user_name, database_name, process_id, connection_from, session_id, session_line_num, command_tag, session_start_time, virtual_transaction_id, transaction_id, error_severity, sql_state_code, message, detail, hint, internal_query, internal_query_pos, context, query, query_pos, location, application_name) FROM STDIN WITH NULL AS 'NULL' QUOTE " CSV; |
| 1870600 | 29/04/2020, 10:15:06 | agent1    | pem           | 64373      | 5ea8fbc6.fb75 | 0              | 127.0.0.1:55512 | COMMIT  | duration: 0.572 ms statement: END;  |
| 1870599 | 29/04/2020, 10:15:06 | agent1    | pem           | 64373      | 5ea8fbc6.fb75 | 5002110        | 127.0.0.1:55512 | UPDATE  | duration: 0.129 ms statement: UPDATE pem.log_configuration SET (last_read_filename, file_offset) = ('/var/lib/pgsql/12/data/log/postgresql-2020-04-29_004252.csv', 3135795) WHERE server_id = 1;  |
| 1870598 | 29/04/2020, 10:15:06 | agent1    | pem           | 64373      | 5ea8fbc6.fb75 | 5002110        | 127.0.0.1:55512 | COPY    | duration: 0.289 ms statement: BEGIN;COPY pemdata.server_logs(server_id, log_time, user_name, database_name, process_id, connection_from, session_id, session_line_num, command_tag, session_start_time, virtual_transaction_id, transaction_id, error_severity, sql_state_code, message, detail, hint, internal_query, internal_query_pos, context, query, query_pos, location, application_name) FROM STDIN WITH NULL AS 'NULL' QUOTE " CSV; |
| 1870597 | 29/04/2020, 10:15:06 | agent1    | pem           | 64373      | 5ea8fbc6.fb75 | 0              | 127.0.0.1:55512 | COMMIT  | duration: 0.521 ms statement: END;  |
| 1870596 | 29/04/2020, 10:15:06 | agent1    | pem           | 64373      | 5ea8fbc6.fb75 | 5002109        | 127.0.0.1:55512 | UPDATE  | duration: 0.132 ms statement: UPDATE pem.log_configuration SET (last_read_filename, file_offset) = ('/var/lib/pgsql/12/data/log/postgresql-2020-04-29_004252.csv', 3134305) WHERE server_id = 1;  |
| 1870595 | 29/04/2020, 10:15:06 | agent1    | pem           | 64373      | 5ea8fbc6.fb75 | 5002109        | 127.0.0.1:55512 | COPY    | duration: 0.200 ms statement: BEGIN;COPY pemdata.server_logs(server_id, log_time, user_name, database_name, process_id, connection_from, session_id, session_line_num, command_tag, session_start_time, virtual_transaction_id, transaction_id, error_severity, sql_state_code, message, detail, hint, internal_query, internal_query_pos, context, query, query_pos, location, application_name) FROM STDIN WITH NULL AS 'NULL' QUOTE " CSV; |
| 1870594 | 29/04/2020, 10:15:06 | agent1    | pem           | 64373      | 5ea8fbc6.fb75 | 0              | 127.0.0.1:55512 | COMMIT  | duration: 0.282 ms statement: END;  |

The header information on the **Server Log Analysis** dashboard displays the date and time that the server was started, the date and time that the page was last updated, and the current number of triggered alerts.

The **Server Log** table displays the contents of the log files that are stored on the PEM server. For content to be displayed, you must check the box next to **Import logs to PEM** when using Log Manager to configure logging for the server.

Entries are displayed in chronological order, most-recent log entries first. Use the scroll bars to navigate through the log entries, or to view columns that are off of the display.

Headings at the top of the server log table identify the information stored in each column:

- The **Id** column identifies the PEM agent that monitors the server that initiated the recorded transaction.
- The **Server** column identifies the server that initiated the recorded transaction.
- The **Timestamp** column displays the date and time that the log entry was made.
- The **User Name** column displays the name of the user that executed the recorded transaction.
- The **Database Name** column displays the name of the database on which the recorded transaction was executed.
- The **Process ID** column displays the identifier of the process that executed the recorded transaction.
- The **Session ID** column displays the identifier of the session in which the transaction was executed.
- The **Transaction ID** column displays the transaction identifier.
- The **Connection From** column displays the host name or IP address from which the client session connected.
- The **Command** column displays the type of command executed.
- The **Message** column displays the transaction message.

Click **Show Filters** to display a panel that you can use to filter the audit log entries that are shown in the table below; click on **Hide Filters** to close the panel.



Properties SQL Statistics Dependencies Dependents Monitoring **Server Log**

Postgres Enterprise Manager Host > Postgres Enterprise Manager Server > **Server Log**

Object Type Server Status UP (Since : 27/04/2020, 15:47:09) Generated On 29/04/2020, 10:23:01 No. of alerts 6 (Acknowledged : 0)

Server Logs

Start: 2020-04-29 09:22:10 End: 2020-04-29 10:22:21 User: postgres Database: pem Command type:

Use the fields within the filter definition box to describe a selection criteria that PEM will use to select a subset of a report for display:

- Use the date and time selectors in the **From** field to specify a starting date and time for the displayed log entries.
- Use the date and time selectors in the **To** field to specify an ending date and time for the displayed log entries.
- Enter a username in the **Username** field to show log entries for the specified user only.
- Enter a database name in the **Database** field to show log entries for the specified database only.
- Enter a command type (for example; 'SELECT', 'authentication' or 'idle') in the **Command type** field to show log entries of that type only.

When you've described the criteria by which you wish to filter the audit logs, click **Filter** to display the filtered server log in the lower portion of the **Server Log Analysis** dashboard. Click the **Hide Filters** label to close the filter definition box.

### 34.4.1.13 The Session Activity Analysis Dashboard

The Session Activity Analysis dashboard provides information about the session workload and lock activity for the selected server:

Properties SQL Statistics Dependencies Dependents Monitoring **Session Activity**

Postgres Enterprise Manager Host > Postgres Enterprise Manager Server > **Session Activity**

Object Type Server Status UP (Since : 27/04/2020, 15:47:09) Generated On 29/04/2020, 10:32:26 No. of alerts 6 (Acknowledged : 0)

Session Activity

Work Load

| Session ID | User Name | Source             | Database Name | Waiting? | Backend Start                    | Transaction Start                | Query Start                      | Memory Usage | Swap Usage | CPU Usage | IO Reads (# bytes) | IO Writes (# bytes) |
|------------|-----------|--------------------|---------------|----------|----------------------------------|----------------------------------|----------------------------------|--------------|------------|-----------|--------------------|---------------------|
| 72556      | postgres  | 127.0.0.1:38696    | postgres      | Yes      | 2020-04-29 10:17:30.901977+05:30 |                                  | 2020-04-29 10:24:02.37053+05:30  | 5.277        | 2.816      | 0.000     | 171464.000         |                     |
| 93857      | postgres  | 127.0.0.1:44690    | postgres      | No       | 2020-04-29 10:24:21.36282+05:30  | 2020-04-29 10:24:21.373882+05:30 | 2020-04-29 10:24:21.373882+05:30 | 7.637        | 2.816      | 0.000     | 171463.000         |                     |
| 37325      | postgres  | 127.0.0.1:51258    | pem           | Yes      | 2020-04-27 16:17:42.962509+05:30 |                                  | 2020-04-28 16:46:59.926278+05:30 | 27.359       | 2.832      | 0.000     | 669633.000         |                     |
| 80541      | postgres  | 127.0.0.1:51876    | pem           | Yes      | 2020-04-27 15:47:11.693572+05:30 |                                  | 2020-04-29 10:24:08.186452+05:30 | 27.484       | 3.254      | 0.000     | 964515.000         |                     |
| 80909      | agent1    | 127.0.0.1:51886    | pem           | Yes      | 2020-04-27 15:47:19.122199+05:30 |                                  | 2020-04-29 10:24:11.98416+05:30  | 36.637       | 3.605      | 0.000     | 8018643.000        | 64                  |
| 81830      | agent1    | 127.0.0.1:51918    | pem           | Yes      | 2020-04-27 15:47:39.499742+05:30 |                                  | 2020-04-29 10:24:17.65002+05:30  | 5.402        | 3.480      | 0.000     | 275595.000         |                     |
| 81831      | agent1    | 127.0.0.1:51920    | pem           | Yes      | 2020-04-27 15:47:39.515256+05:30 |                                  | 2020-04-29 10:24:17.649713+05:30 | 6.051        | 2.863      | 0.000     | 275595.000         |                     |
| 17375      | postgres  | 127.0.0.1:55154    | pem           | Yes      | 2020-04-28 16:45:57.218333+05:30 |                                  | 2020-04-29 08:19:05.921713+05:30 | 28.441       | 2.805      | 0.000     | 546717.000         |                     |
| 64373      | agent1    | 127.0.0.1:55512    | pem           | Yes      | 2020-04-29 09:30:06.515126+05:30 |                                  | 2020-04-29 10:24:21.372505+05:30 | 142.891      | 2.785      | 20.500    | 43399177.000       | 47                  |
| 58913      | postgres  | 192.168.1.13:54699 | pem           | Yes      | 2020-04-28 13:10:48.241955+05:30 |                                  | 2020-04-28 19:47:55.297426+05:30 | 43.063       | 5.043      | 0.000     | 5986609.000        |                     |
| 58914      | postgres  | 192.168.1.13:54710 | pem           | Yes      | 2020-04-28 13:10:48.392445+05:30 |                                  | 2020-04-28 19:47:55.714573+05:30 | 34.492       | 5.387      | 0.000     | 792881.000         |                     |
| 115301     | postgres  | 192.168.1.13:62921 | pem           | Yes      | 2020-04-29 09:02:44.994761+05:30 |                                  | 2020-04-29 09:02:45.575466+05:30 | 13.211       | 2.801      | 0.000     | 2996804.000        |                     |

The Session Activity Analysis dashboard header displays the date and time that the server was started, the date and time that the dashboard was last

updated and the number of current alerts for the server.

Use parameters on the [PEM Server Configurations](#) dialog to specify the auto-refresh rate for the [Session Activity Analysis](#) dashboard. To access the [Server Configuration](#) dialog, select [Server Configuration...](#) from the PEM client [Management](#) menu.

The [Session Workload](#) table provides information about the current session workload for the server. Click a column heading to sort the table data by the selected column; click the heading a second time to reverse the sort order. The Session Workload table displays the following information:

- The [Session ID](#) column displays the process identifier for the session.
- The [User Name](#) column displays the (role) name of the user that established the client connection to the server.
- The [Source](#) column displays the IP address and port number of the client.
- The [Database Name](#) column displays the name of the database to which the client is connected.
- The [Waiting](#) column displays [Yes](#) if the session is waiting for a lock; [No](#) if the session is not waiting for a lock.
- The [Backend Start](#) column displays the date and time that the client established a connection to the server.
- The [Transaction Start](#) column displays the date and time that the current transaction started, if applicable.
- The [Query Start](#) column displays the date and time that the current query started, if applicable.
- The [Memory Usage](#) column displays the amount of memory used by the session; this column is not displayed if the server is remotely monitored.
- The [Swap Usage](#) column displays the amount of swap space used by the session; this column is not displayed if the server is remotely monitored.
- The [CPU Usage](#) column displays the amount of CPU resources used by the session; this column is not displayed if the server is remotely monitored.
- The [IO Reads \(#bytes\)](#) column displays the number of bytes used by the session while reading transactions; this column is not displayed if the server is remotely monitored.
- The [IO Writes \(#bytes\)](#) column displays the number of bytes used by the session while writing transactions; this column is not displayed if the server is remotely monitored.

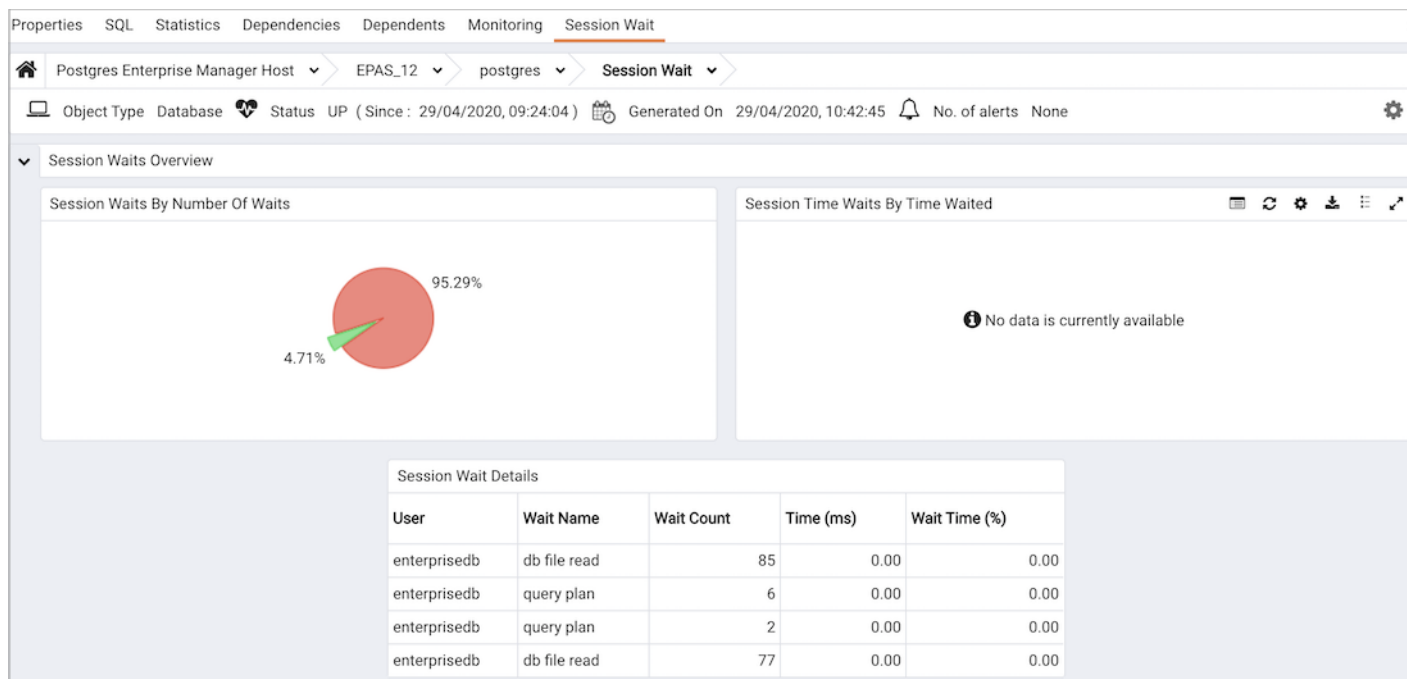
The [Session Lock Activity](#) table displays a list of locks held by processes on the server. Click a column heading to sort the table data by the selected column; click the heading a second time to reverse the sort order. The Session Lock Activity table displays the following information:

- The [Session ID](#) column displays the process ID for the session.
- The [User Name](#) column displays the name of the user holding (or waiting for) the lock.
- The [Source](#) column displays the IP address and port number of the client.
- The [Database Name](#) column displays the name of the database to which the client is connected.
- The [Blocked](#) column indicates if the lock request is blocked by another lock.
- The [Blocked By](#) column specifies the session ID of the session that is holding the lock.
- The [Lock Type](#) column displays the type of lock that is held by the client. Lock Type may be:
  - [advisory](#) - a user-defined lock created by `pg_advisory_lock()` or `pg_advisory_lock_shared()`
  - [extend](#) - a lock held while extending a table or index
  - [object](#) - a lock held on a database object
  - [page](#) - a lock held on a page (within the shared buffer cache)
  - [relation](#) - a lock held on the metadata describing a table, view, or sequence (to prevent another session from altering the table, view, or sequence)
  - [transactionid](#) - a lock held on a transaction ID (one session typically waits for another transaction to complete by waiting on the other session's transaction ID)
  - [tuple](#) - lock held on a tuple (typically, a tuple which has been inserted, updated, or deleted, but not yet committed)
  - [userlock](#) - a user-defined lock created with the `LOCK` statement
  - [virtualxid](#) - a lock identified by a virtual transaction ID.
- The [Object ID](#) column displays the OID of the relation, or NULL if the object is not a relation (of part of a relation).
- The [Mode](#) column displays the name of the lock mode held (or sought) by the process.
- The [Transaction Start](#) column displays the date and time that the transaction started.

### 34.4.1.14 The Session Waits Analysis Dashboard

The [Session Wait Analysis](#) dashboard provides an overview of the current DRITA wait events for an Advanced Server session. For more information about DRITA wait events, please see the EDB Postgres Advanced Server Guide.





Use parameters on the [PEM Server Configurations](#) dialog to specify the auto-refresh rate for the [Alerts](#) dashboard. To access the [Server Configuration](#) dialog, select [Server Configuration...](#) from the PEM web interface [Management](#) menu.

The Session Wait Analysis dashboard header displays the date and time that the server started, the date and time that the dashboard was last updated, and the number of alerts currently triggered for the specified database (and monitored objects that reside within that database).

The [Session Waits Overview](#) displays statistics gathered by the most recent execution of the PEM probe:

- The [Session Waits By Number Of Waits](#) pie chart displays the 5 most frequently encountered wait events, per Advanced Server session. For more information about the events that can cause a wait event, see the EDB Postgres Advanced Server Guide.
- The [Session Waits By Time Waited](#) pie chart displays the 5 wait events that consume the most time, per Advanced Server session. To gather and display data in the [Session Time Waits by Time Waited](#) pie chart, you must modify the `postgresql.conf` file for the monitored server, setting `timed_statistics = on`, and restart the server. Please note that this will cause server performance to degrade. For more information about using Advanced Server DRITA timers and the events that can cause a wait event, please see the EDB Postgres Advanced Server Guide.

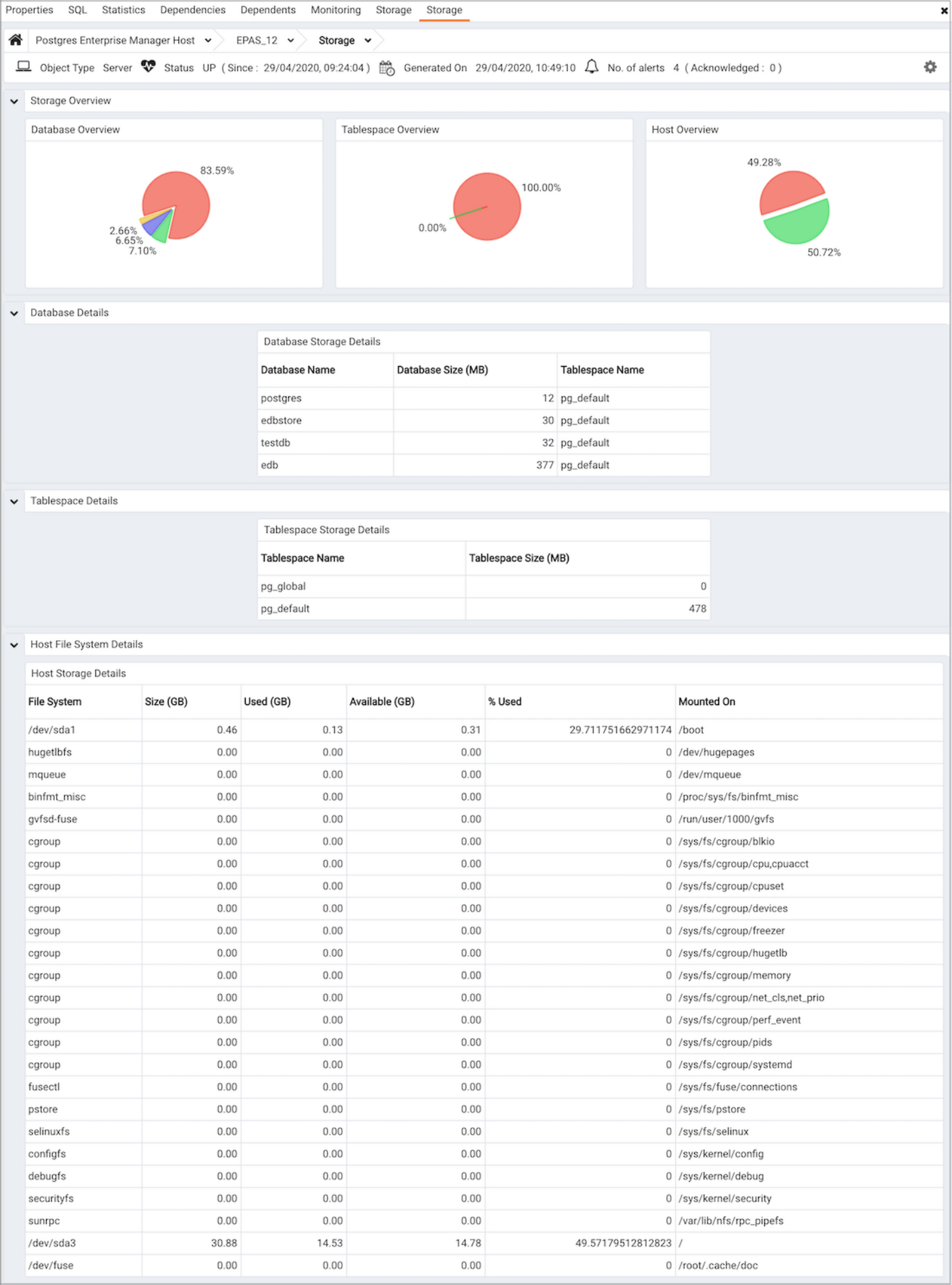
The [Session Waits Details](#) table lists the current system wait events for the selected database. Click a column heading to sort the table by the column data; click again to reverse the sort order. The table displays:

- The [User](#) column displays the name of the user that encountered the wait.
- The [Wait Name](#) column displays the name of the of wait event.
- The [Wait Count](#) column displays the total number of waits encountered by the user.
- The [Time \(ms\)](#) displays the number of milliseconds that the user waited for the specified event.
- The [Wait Time \(%\)](#) column displays the percentage of the total wait time consumed by the specified wait event.

To gather and display data in the Time (ms) and Wait Time (%) columns, you must modify the `postgresql.conf` file for the monitored server, setting `timed_statistics = on`, and restart the server. Please note that this will cause server performance to degrade. For more information about using Advanced Server DRITA timers, please see the EDB Postgres Advanced Server Guide.

### 34.4.1.15 The Storage Analysis Dashboard

The [Storage Analysis](#) dashboard provides information about the size of objects stored on the server and about available storage space on the server.



Use parameters on the [PEM Server Configurations](#) dialog to specify the auto-refresh rate for the dashboard. To access the [Server Configuration](#)

dialog, select **Server Configuration...** from the PEM web interface **Management** menu.

The Storage Analysis dashboard header displays the date and time that the PEM server started, the date and time that the dashboard was most recently updated, and the number of triggered alerts on objects monitored by the PEM server.

The Storage Overview section displays information about the size of databases, tablespaces and the host:

- The **Database Overview** pie chart shows the relative size of monitored databases stored on the server. The key (located below the chart) matches the database name to the respective color on the chart.
- The **Tablespace Overview** pie chart shows the relative size of tablespaces on the server. The key (located below the chart) matches the tablespace name to the respective color on the chart.
- The **Host Overview** pie chart represents the amount of used and free storage space on the server as of the last probe execution.

The **Database Details** table displays the size of each database stored on the server. Click a column heading to sort the table by the specified column; click again to reverse the sort order.

- The **Database Name** column displays the name of the database.
- The **Database Size (MB)** column displays the size of the database in megabytes.
- The **Tablespace Name** column displays the name of the default tablespace assigned to the database.

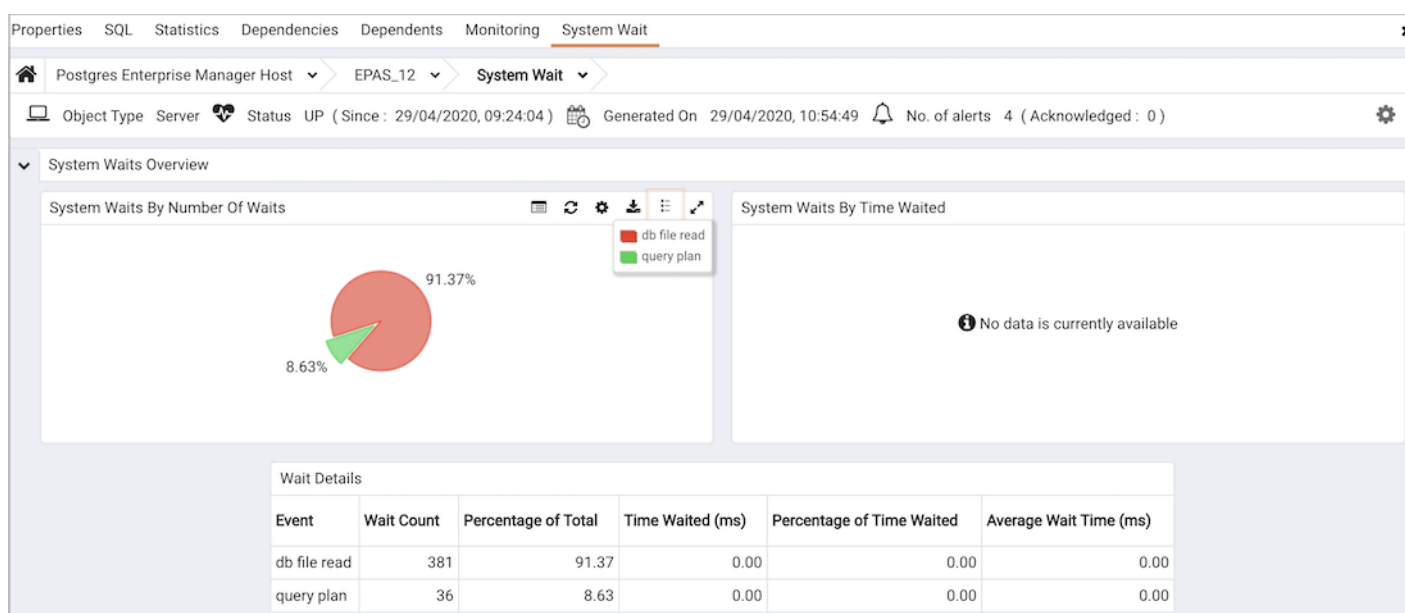
The **Tablespace Details** table lists the name and size (in megabytes) of each tablespace defined for the server. Click a column heading to sort the table by the specified column; click again to reverse the sort order.

The **Host File System Details** table displays information about the file systems that reside on the system that hosts the PEM server:

- The **File System** column displays the name of the file system.
- The **Size (GB)** column displays the size of the file system in megabytes.
- The **Used (GB)** column displays the amount of the file system that is currently storing information.
- The **Available (GB)** column displays the amount of space available on the file system.
- The **% Used** column displays the percentage of the total storage space in use.
- The **Mounted On** column displays the directory on which the file system is mounted.

### 34.4.1.16 The System Wait Analysis Dashboard

The System Wait Analysis dashboard provides an overview of the current DRITA wait events for an Advanced Server database. For more information about DRITA wait events, please see the EDB Postgres Advanced Server Guide.



Use parameters on the [PEM Server Configurations](#) dialog to specify the auto-refresh rate for the [Alerts](#) dashboard. To access the [Server Configuration](#) dialog, select [Server Configuration...](#) from the PEM web interface [Management](#) menu.

The System Waits Analysis dashboard header displays the date and time that the server started, the date and time that the System Waits Analysis dashboard was last updated, and the number of alerts currently triggered for the specified database (and monitored objects that reside within that database).

The [System Waits Overview](#) displays statistics gathered by the most recent execution of the PEM probe:

- The [System Waits by Number of Waits](#) pie chart displays the 5 most frequently encountered wait events for the selected Advanced Server server. For more information about the events that can cause a wait event, see the EDB Postgres Advanced Server Guide.
- The [System Waits by Time Waited](#) pie chart displays the 5 wait events that consume the most time for the selected Advanced Server server. To gather and display data in the [System Waits by Time Waited](#) pie chart, you must modify the [postgresql.conf](#) file for the monitored server, setting `timed_statistics = on`, and restart the server. Please note that this will cause server performance to degrade. For more information about using Advanced Server DRITA timers, please see the EDB Postgres Advanced Server Guide.

The [System Waits Details](#) table lists the current system wait events for the selected server. Click a column heading to sort the table by the column data; click again to reverse the sort order. The table displays:

- The [Event](#) column displays the name of the wait event.
- The [Wait Count](#) column contains the number of times that the wait event occurred.
- The [Percent of Total](#) column displays the percentage of the total wait count consumed by this event.
- The [Time Waited \(ms\)](#) displays the number of milliseconds that the server waited for the event.
- The [Percent of Time Waited](#) displays the percentage of the total wait time consumed by this event.
- The [Average Wait Time \(ms\)](#) column displays the average wait time for this event.

To gather and display data in the [Time Waited \(ms\)](#) and [Percent of Time Waited](#) columns, you must modify the [postgresql.conf](#) file for the monitored server, setting `timed_statistics = on`, and restart the server. Please note that this will cause server performance to degrade. For more information about using Advanced Server DRITA timers, please see the EDB Postgres Advanced Server Guide.

### 34.4.1.17 The Streaming Replication Analysis Dashboard

The [Streaming Replication Analysis](#) Dashboard displays statistical information about WAL activity for a monitored server. By default, replication probes are disabled; to view the [Streaming Replication Analysis](#) dashboard, you must enable probes on the primary and replica nodes. To enable the probes on the primary node, highlight the name of the primary server in the PEM client [Browser](#) tree control, and select [Manage Probes...](#) from the [Management](#) menu. Use the [Manage Probes](#) tab to enable the following probes:

- Streaming Replication
- WAL Archive Status

To enable the probes on the replica node, highlight the name of the replica server in the PEM client [Browser](#) tree control, and select [Manage Probes...](#) from the [Management](#) menu. Use the [Manage Probes](#) tab to enable the following probe:

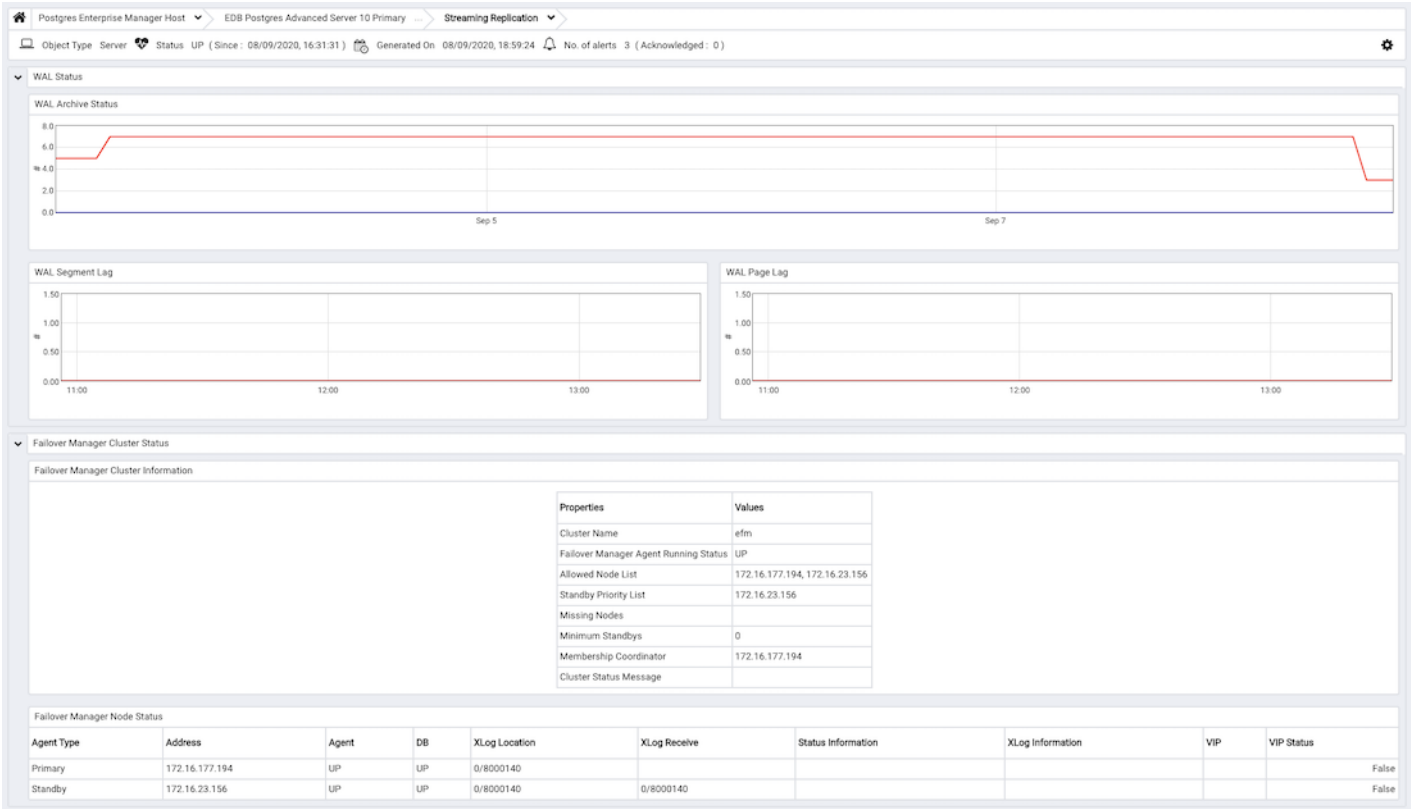
- Streaming Replication Lag Time

Then, to open the [Streaming Replication Analysis](#) dashboard, navigate to the [Monitoring](#) tab, and:

1. Select the name of the agent that monitors the node from the [Agents](#) drop-down menu.
2. Select the name of the monitored server from the [Servers](#) drop-down menu.
3. Select [Streaming Replication Analysis](#) from the [Dashboards](#) drop-down menu.

The [Streaming Replication Analysis](#) dashboard header includes the date and time that the server was last started, the date and time that the page was last updated, and a current count of triggered alerts.

When accessing the [Streaming Replication Analysis](#) dashboard for the primary node of a replication scenario, the dashboard displays information about the write-ahead log activity for the server.



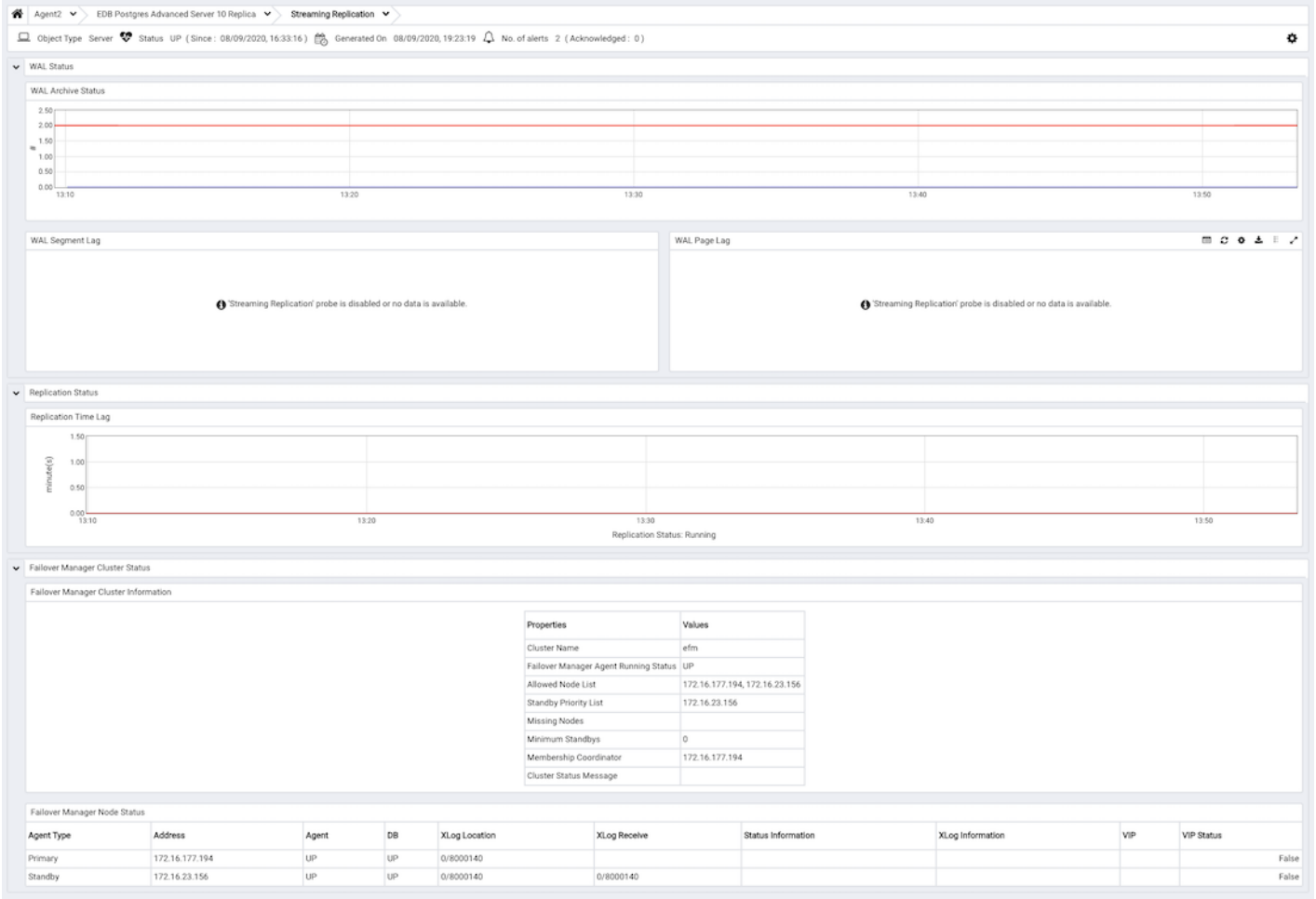
The **WAL Archive Status** graph displays WAL activity; the vertical key on the left side of the graph indicates the archive count; times are displayed across the bottom of the graph.

The **WAL Segment Lag** graph displays the segment lag for the replica nodes that are associated with the selected server. The vertical key on the left side of the graph indicates the archive count. Each node is displayed in a different color on the graph. The **Legend** provides a key to the identity (hostname and port) of each graphed replica node.

The **WAL Page Lag** graph displays the page lag activity for each replica node associated with the selected server. The vertical key on the left side of the graph indicates the page count. Each node is displayed in a different color on the graph. The **Legend** provides a key to the identity (hostname and port) of each graphed replica node.

Monitoring a Replica Node

When accessing the **Streaming Replication Analysis** dashboard for the replica node of a replication scenario, the dashboard displays information about the write-ahead log activity for the server.



The **WAL Archive Status** graph displays WAL activity; the vertical key on the left side of the graph indicates the archive count; times are displayed across the bottom of the graph.

The **WAL Segment Lag** graph displays the segment lag for the replica nodes that are associated with the selected server. The vertical key on the left side of the graph indicates the archive count. Each replica node is displayed in a different color on the graph. The **Legend** provides a key to the identity (hostname and port) of each graphed slave node.

The **WAL Page Lag** graph displays the page lag activity for each replica node associated with the selected server. The vertical key on the left side of the graph indicates the page count. Each node is displayed in a different color on the graph. The **Legend** provides a key to the identity (hostname and port) of each graphed slave node.

The **Replication Time Lag** graph displays the delay between the time that an operation is performed on the primary node of the replication scenario and the time that the operation is written to the replica node. The vertical key on the left side of the graph indicates the replication delay in minutes. Hover your mouse over a point on the graph to display the date and time that corresponds to that coordinate.

A label at the bottom of the dashboard confirms the status of the replication replica.

### Monitoring a Failover Manager Cluster

If you have configured PEM to monitor a **Failover Manager** cluster, the Streaming Replication Analysis dashboard will display tables that provide an overview of the clusters status and configuration, and information about each cluster member. To display cluster information on the Streaming Replication dashboard, you must provide the following information on the **Advanced** tab of the server **Properties** dialog for each node of the cluster:

- Use the **EFM Cluster Name** field to specify the name of the Failover Manager cluster. The cluster name is the prefix of the name of the cluster properties file. For example, if your cluster properties file is named **efm.properties**, your cluster name is **efm**.
- Use the **EFM Installation Path** field to specify the location of the Failover Manager binary file. By default, the Failover Manager binary file is

installed in `/usr/edb/efm-3.1/bin`.

The **Failover Manager Cluster Status** section of the Streaming Replication Analysis dashboard displays information about the monitored cluster:

Failover Manager Cluster Status

Failover Manager Cluster Information

| Properties                            | Values                        |
|---------------------------------------|-------------------------------|
| Cluster Name                          | efm                           |
| Failover Manager Agent Running Status | UP                            |
| Allowed Node List                     | 172.16.177.194, 172.16.23.156 |
| Standby Priority List                 | 172.16.23.156                 |
| Missing Nodes                         |                               |
| Minimum Standbys                      | 0                             |
| Membership Coordinator                | 172.16.177.194                |
| Cluster Status Message                |                               |

Failover Manager Node Status

| Agent Type | Address        | Agent | DB | XLog Location | XLog Receive | Status Information | XLog Information | VIP | VIP Status |
|------------|----------------|-------|----|---------------|--------------|--------------------|------------------|-----|------------|
| Primary    | 172.16.177.194 | UP    | UP | 0/8000140     |              |                    |                  |     | False      |
| Standby    | 172.16.23.156  | UP    | UP | 0/8000140     | 0/8000140    |                    |                  |     | False      |

The **Failover Manager Cluster Information** table provides information about the Failover Manager cluster:

- The **Properties** column displays the name of the cluster property.
- The **Values** column displays the current value of the property.

The **Failover Manager Node Status** table displays information about each node of the Failover Manager cluster:

- The **Agent Type** column displays the type of agent that resides on the node; the possible values are Primary, Replica, Witness, Idle, and Promoting.
- The **Address** column displays the IP address of the node.
- The **Agent** column displays the status of the agent that resides on the node.
- The **DB** column displays the status of the database that resides on the node.
- The **XLog Location** column displays the transaction log location of the database.
- The **Status Information** column displays any error-related information about the node.
- The **XLog Information** column displays any error-related information about the transaction log.
- The **VIP** column displays the VIP address that is associated with the node.
- The **VIP Status** column displays **True** if the VIP is active for the node, **False** if the VIP is not.

34.4.2 Server Configuration

You can use the **Server Configuration** dialogue to modify values of user-configurable parameters that control the behavior of Postgres Enterprise Manager. To access the **Server Configuration** dialog, connect to the PEM server, and select **Server Configuration...** from the **File** menu.

Server Configuration

Search by parameter name

| Parameter                  | Value           | Unit    |
|----------------------------|-----------------|---------|
| audit_log_retention_time   | 30              | days    |
| auto_create_agent_alerts   | <div>True</div> | t/f     |
| auto_create_server_alerts  | <div>True</div> | t/f     |
| bart_log_retention_time    | 30              | days    |
| cm_data_points_per_report  | 50              |         |
| cm_max_end_date_in_years   | 5               | years   |
| dash_alerts_timeout        | 60              | seconds |
| dash_db_comrol_span        | 168             | hours   |
| dash_db_comrol_timeout     | 1800            | seconds |
| dash_db_connovervw_timeout | 300             | seconds |
| dash_db_eventlag_span      | 7               | days    |
| dash_db_eventlag_timeout   | 1800            | seconds |
| dash_db_hottable_rows      | 25              | rows    |
| dash_db_hottable_timeout   | 300             | seconds |
| dash_db_io_span            | 168             | hours   |
| dash db io timeout         | 1800            | seconds |

?

Cancel

Reset

Save

Enter a parameter name in the search box in the upper-right corner of the dialog to locate a specific parameter in the list.

To modify a parameter value, edit the content displayed in the **Value** field to the right of a parameter name. Click the **Save** icon in the upper-right corner of the dialog to save your changes, or click the **Close** button to exit the dialog without applying the changes.

A list of configuration options may be found [here](#).

### 34.4.2.1 Configuration Options

A number of aspects of PEM's behaviour can be controlled using global configuration options. Use the [Server Configuration dialogue](#) to manage Server Options. The configuration parameters used are listed below.

Please note that this list is subject to change.

| Parameter name            | Value/Unit | Description  |
|---------------------------|------------|--|
| audit_log_retention_time  | 30 days    | Specifies the number of days that an audit log will be retained on the PEM server.   |
| auto_create_agent_alerts  | true       | Specifies whether to create default agent level alerts automatically when an agent is registered.  |
| auto_create_server_alerts | true       | Specifies whether to create default server level alerts automatically when a server is bound to an agent.  |
| chart_disable_bullets     | false      | Enable/disable bullets on line charts on dashboards and Capacity Manager reports.  |
| cm_data_points_per_report | 50         | Specifies the number of data points to plot on charts on Capacity Manager reports.   |
| cm_max_end_date_in_years  | 5 years    | Specifies the maximum amount of time that the Capacity Manager will extrapolate data for. Ensures that threshold-based end dates of on reports do not get extrapolated indefinitely. |
| dash_alerts_timeout       | 60 seconds | Specifies the number of seconds after which the components of the Alerts dashboard are auto-refreshed.   |



| Parameter name               | Value/Unit   | Description  |
|------------------------------|--------------|--|
| dash_db_comrol_span          | 7 days       | Specifies the number of days worth of data to plot on the Commit/Rollback Analysis chart on the Database Analysis dashboard and Server Analysis dashboard.                                   |
| dash_db_comrol_timeout       | 1800 seconds | Specifies the number of seconds after which the Commits/Rollbacks line chart is auto-refreshed on the Database Analysis dashboard and Server Analysis dashboard.                             |
| dash_db_connovervw_timeout   | 300 seconds  | Specifies the number of seconds after which the Connection Overview pie chart is auto-refreshed in the Database Analysis dashboard.  |
| dash_db_eventlag_span        | 7 days       | Specifies the number of days worth of data to plot on the Number of Events Lag chart for slony replication on the Database Analysis dashboard.   |
| dash_db_eventlag_timeout     | 1800 seconds | Specifies the number of seconds after which the Number of Events Lag line chart for slony replication is auto-refreshed on the Database Analysis dashboard.                                  |
| dash_db_hottable_rows        | 25 rows      | Specifies the number of rows to show on the HOT Table Analysis table on the Database Analysis dashboard.   |
| dash_db_hottable_timeout     | 300 seconds  | Specifies the number of seconds after which the Hot Tables table is auto-refreshed in the Database Analysis dashboard.   |
| dash_db_io_span              | 7 days       | Specifies the number of days worth of data to plot on the Database I/O Analysis chart on the Database Analysis dashboard and I/O Analysis dashboard.   |
| dash_db_io_timeout           | 1800 seconds | Specifies the number of seconds after which the Database I/O line chart is auto-refreshed on the Database Analysis dashboard and I/O Analysis dashboard.                                     |
| dash_db_rowact_span          | 7 days       | Specifies the number of days worth of data to plot on the Row Activity Analysis chart on the Database Analysis dashboard, the I/O Analysis dashboard, and the Server Analysis dashboard.     |
| dash_db_rowact_timeout       | 1800 seconds | Specifies the number of seconds after which the Row Activity line chart is auto-refreshed on the Database Analysis dashboard, the I/O Analysis dashboard, and the Server Analysis dashboard. |
| dash_db_storage_timeout      | 300 seconds  | Specifies the number of seconds after which the Storage bar chart is auto-refreshed in the Database Analysis dashboard.  |
| dash_db_timelag_span         | 7 days       | Specifies the number of days worth of data to plot on the Time Lag chart for Slony replication on the Database Analysis dashboard.   |
| dash_db_timelag_timeout      | 1800 seconds | Specifies the number of seconds after which the Time Lag line chart for slony replication is auto-refreshed on the Database Analysis dashboard.  |
| dash_db_useract_span         | 7 days       | Specifies the number of days worth of data to plot on the User Activity Analysis chart on the Database Analysis dashboard.   |
| dash_db_useract_timeout      | 1800 seconds | Specifies the number of seconds after which the User Activity line chart is auto-refreshed in the Database Analysis dashboard.   |
| dash_efm_timeout             | 300 seconds  | Specifies the number of seconds after which the Failover Manager Node Status and Failover Manager Cluster Info line chart is auto-refreshed on the Streaming Replication dashboard.          |
| dash_global_overview_timeout | 30 seconds   | Specifies the number of seconds after which the components of the Global Overview dashboard are auto-refreshed.  |
| dash_header_timeout          | 60 seconds   | Specifies the number of seconds after which the information on the header of all the dashboards are auto-refreshed.  |
| dash_io_chkpt_span           | 7 days       | Specifies the number of days worth of data to plot on the Checkpoints chart on the I/O Analysis dashboard.   |
| dash_io_chkpt_timeout        | 1800 seconds | Specifies the number of seconds after which the Checkpoints line chart is auto-refreshed on the I/O Analysis dashboard.  |
| dash_io_hotindx_timeout      | 300 seconds  | Specifies the number of seconds after which the Hot Indexes bar chart is auto-refreshed on the I/O Analysis dashboard.   |
| dash_io_hottbl_timeout       | 300 seconds  | Specifies the number of seconds after which the Hot Tables bar chart is auto-refreshed on the I/O Analysis dashboard.  |
| dash_io_index_objectio_rows  | 25 rows      | Specifies the number of rows displayed on the Index Activity table on the I/O Analysis dashboard and the Object Activity Analysis dashboard.   |

| Parameter name                       | Value/Unit   | Description   |
|--------------------------------------|--------------|---|
| dash_io_index_objectio_timeout       | 60 seconds   | Specifies the number of seconds after which the Index Activity table is auto-refreshed on the I/O Analysis dashboard and the Object Activity Analysis dashboard.      |
| dash_io_objectio_rows                | 25 rows      | Specifies the number of rows displayed in the Object I/O Details table on the I/O Analysis dashboard and Object Activity Analysis dashboard.                          |
| dash_io_objectio_timeout             | 300 seconds  | Specifies the number of seconds after which the Object I/O Details table is auto-refreshed on the I/O Analysis dashboard and Object Activity Analysis dashboard.      |
| dash_memory_hostmemact_span          | 7 days       | Specifies the number of days worth of data to plot on the Host Memory Activity Analysis chart on the Memory Analysis dashboard.                                       |
| dash_memory_hostmemact_timeout       | 1800 seconds | Specifies the number of seconds after which the Host Memory Activity line chart is auto-refreshed on the Memory Analysis dashboard.                                   |
| dash_memory_hostmemconf_timeout      | 300 seconds  | Specifies the number of seconds after which the Host Memory Configuration pie chart is auto-refreshed on the Memory Analysis dashboard and Server Analysis dashboard. |
| dash_memory_servmemact_span          | 7 days       | Specifies the number of days worth of data to plot on the server Memory Activity Analysis chart on the Memory Analysis dashboard.                                     |
| dash_memory_servmemact_timeout       | 1800 seconds | Specifies the number of seconds after which the Server Memory Activity line chart is auto-refreshed on the Memory Analysis dashboard.                                 |
| dash_memory_servmemconf_timeout      | 300 seconds  | Specifies the number of seconds after which the Server Memory Configuration pie chart is auto-refreshed on the Memory Analysis dashboard.                             |
| dash_objectact_objstorage_rows       | 15 rows      | Specifies the number of rows to show on the Object Storage table on the Object Activity Analysis dashboard.   |
| dash_objectact_objstorage_timeout    | 300 seconds  | Specifies the number of seconds after which the Object Storage table is auto-refreshed in the Object Activity Analysis dashboard.                                     |
| dash_objectact_objtopindexes_timeout | 300 seconds  | Specifies the number of seconds after which the Top 5 Largest Indexes bar chart is auto-refreshed in the Object Activity Analysis dashboard.                          |
| dash_objectact_objtoptables_timeout  | 300 seconds  | Specifies the number of seconds after which the Top 5 Largest Tables bar chart is auto-refreshed in the Object Activity Analysis dashboard.                           |
| dash_os_cpu_span                     | 7 days       | Specifies the number of days worth of data to plot on the CPU chart on the Operating System Analysis dashboard.   |
| dash_os_cpu_timeout                  | 1800 seconds | Specifies the number of seconds after which the CPU line chart is auto-refreshed on the Operating System Analysis dashboard.  |
| dash_os_data_span                    | 7 days       | Specifies the number of days worth of data to plot on the I/O line chart on the Operating System Analysis dashboard.  |
| dash_os_disk_span                    | 7 days       | Specifies the number of days worth of data to plot on the Utilisation chart on the Operating System Analysis dashboard.   |
| dash_os_hostfs_timeout               | 1800 seconds | Specifies the number of seconds after which the Host File System Details table is auto-refreshed on the Operating System Analysis dashboard.                          |
| dash_os_io_timeout                   | 1800 seconds | Specifies the number of seconds after which the I/O line chart is auto-refreshed on the Operating System Analysis dashboard.  |
| dash_os_memory_span                  | 7 days       | Specifies the number of days worth of data to plot on the Memory chart on the Operating System Analysis dashboard.  |
| dash_os_memory_timeout               | 1800 seconds | Specifies the number of seconds after which the Memory line chart is auto-refreshed on the Operating System Analysis dashboard.                                       |
| dash_os_packet_span                  | 7 days       | Specifies the number of days worth of data to plot on the Packet chart on the Operating System Analysis dashboard.  |
| dash_os_packet_timeout               | 1800 seconds | Specifies the number of seconds after which the Network Packets line chart is auto-refreshed on the Operating System Analysis dashboard.                              |
| dash_os_process_span                 | 7 days       | Specifies the number of days worth of data to plot on the Process chart on the Operating System Analysis dashboard.   |
| dash_os_process_timeout              | 1800 seconds | Specifies the number of seconds after which the Process line chart is auto-refreshed on the Operating System Analysis dashboard.                                      |
| dash_os_storage_timeout              | 1800 seconds | Specifies the number of seconds after which the Storage pie chart is auto-refreshed on the Operating System Analysis dashboard.                                       |
| dash_os_traffic_span                 | 7 days       | Specifies the number of days worth of data to plot on the Traffic chart on the Operating System Analysis dashboard.   |

| Parameter name                       | Value/Unit   | Description   |
|--------------------------------------|--------------|---|
| dash_os_traffic_timeout              | 1800 seconds | Specifies the number of seconds after which the Traffic line chart is auto-refreshed on the Operating System Analysis dashboard.            |
| dash_os_util_timeout                 | 1800 seconds | Specifies the number of seconds after which the Utilisation line chart is auto-refreshed on the Operating System Analysis dashboard.        |
| dash_probe_log_timeout               | 300 seconds  | Specifies the number of seconds after which the Probe Log table is auto-refreshed on the Probe Log Analysis dashboard.                      |
| dash_replication_archivestat_span    | 7 days       | Specifies the number of days worth of data to plot on the WAL Archive Status chart on the Streaming Replication Analysis dashboard.         |
| dash_replication_archivestat_timeout | 1800 seconds | Specifies the number of seconds after which the WAL Archive Status line chart is auto-refreshed on the Streaming Replication dashboard.     |
| dash_replication_pagelag_span        | 7 days       | Specifies the number of days worth of data to plot on the WAL Lag Pages chart on the Streaming Replication dashboard.                       |
| dash_replication_pagelag_timeout     | 1800 seconds | Specifies the number of seconds after which the WAL Lag Pages line chart is auto-refreshed on the Streaming Replication dashboard.          |
| dash_replication_segmentlag_span     | 7 days       | Specifies the number of days worth of data to plot on the WAL Lag Segments chart on the Streaming Replication dashboard.                    |
| dash_replication_segmentlag_timeout  | 1800 seconds | Specifies the number of seconds after which the WAL Lag Segments line chart is auto-refreshed on the Streaming Replication dashboard.       |
| dash_replication_timelag_span        | 7 days       | Specifies the number of days worth of data to plot on the Replication Lag Time chart on the Streaming Replication dashboard.                |
| dash_replication_timelag_timeout     | 1800 seconds | Specifies the number of seconds after which the Replication Lag Time line chart is auto-refreshed on the Streaming Replication dashboard.   |
| dash_server_buffers_written          | 168 hours    | Specifies the number of days worth of data to plot on the Background Writer Statistics chart on the Server Analysis dashboard.              |
| dash_server_buffers_written_timeout  | 300 seconds  | Specifies the number of seconds after which the Background Writer Statistics line chart is auto-refreshed on the Server Analysis dashboard. |
| dash_server_connovervw_timeout       | 300 seconds  | Specifies the number of seconds after which the Connection Overview pie chart is auto-refreshed in the Server Analysis dashboard.           |
| dash_server_database_timeout         | 300 seconds  | Specifies the number of seconds after which the Databases table is auto-refreshed in the Server Analysis dashboard.                         |
| dash_server_dbsize_span              | 7 days       | Specifies the number of days worth of data to plot on the Database Size Analysis chart on the Server Analysis dashboard.                    |
| dash_server_dbsize_timeout           | 1800 seconds | Specifies the number of seconds after which the Database Size line chart is auto-refreshed in the Server Analysis dashboard.                |
| dash_server_disk_timeout             | 1800 seconds | Specifies the number of seconds after which the Disk line chart is auto-refreshed in the Server Analysis dashboard.                         |
| dash_server_global_span              | 7 days       | Specifies the number of days worth of data to plot on the Disk line chart on the Server Analysis dashboard.                                 |
| dash_server_sharedbuff_span          | 7 days       | Specifies the number of days worth of data to plot on the Shared Buffer chart on the Server Analysis dashboard.                             |
| dash_server_sharedbuff_timeout       | 1800 seconds | Specifies the number of seconds after which the Shared Buffers line chart is auto-refreshed in the Server Analysis dashboard.               |
| dash_server_tabspacesize_span        | 7 days       | Specifies the number of days worth of data to plot on the Tablespace Size chart on the Server Analysis dashboard.                           |
| dash_server_tabspacesize_timeout     | 1800 seconds | Specifies the number of seconds after which the Tablespace Size line chart is auto-refreshed in the Server Analysis dashboard.              |
| dash_server_useract_span             | 7 days       | Specifies the number of days worth of data to plot on the User Activity chart on the Server Analysis dashboard.                             |
| dash_server_useract_timeout          | 1800 seconds | Specifies the number of seconds after which the User Activity line chart is auto-refreshed in the Server Analysis dashboard.                |
| dash_sessact_lockact_timeout         | 300 seconds  | Specifies the number of seconds after which the Session Lock Activity table is auto-refreshed in the Session Activity Analysis dashboard.   |
| dash_sessact_workload_timeout        | 300 seconds  | Specifies the number of seconds after which the Session Workload table is auto-refreshed in the Session Activity Analysis dashboard.        |

| Parameter name                    | Value/Unit  | Description   |
|-----------------------------------|-------------|---|
| dash_sess_waits_nowaits_timeout   | 300 seconds | Specifies the number of seconds after which the Session Waits By Number Of Waits pie chart is auto-refreshed in the Session Waits Analysis dashboard.   |
| dash_sess_waits_timewait_timeout  | 300 seconds | Specifies the number of seconds after which the Session Waits By Time Waited pie chart is auto-refreshed in the Session Waits Analysis dashboard.   |
| dash_sess_waits_waitdtl_timeout   | 300 seconds | Specifies the number of seconds after which the Session Waits Details table is auto-refreshed in the Session Waits Analysis dashboard.  |
| dash_storage_dbdtls_timeout       | 300 seconds | Specifies the number of seconds after which the Database Details table is auto-refreshed in the Storage Analysis dashboard.   |
| dash_storage_dbovervw_timeout     | 300 seconds | Specifies the number of seconds after which the Database Overview pie chart is auto-refreshed in the Storage Analysis dashboard.  |
| dash_storage_hostdtls_timeout     | 300 seconds | Specifies the number of seconds after which the Host Details table is auto-refreshed in the Storage Analysis dashboard.   |
| dash_storage_hostovervw_timeout   | 300 seconds | Specifies the number of seconds after which the Host Overview pie chart is auto-refreshed in the Storage Analysis dashboard.  |
| dash_storage_tblspcdtls_timeout   | 300 seconds | Specifies the number of seconds after which the Tablespace Details table is auto-refreshed in the Storage Analysis dashboard.   |
| dash_storage_tblspcovervw_timeout | 300 seconds | Specifies the number of seconds after which the Tablespace Overview pie chart is auto-refreshed in the Storage Analysis dashboard.  |
| dash_sys_waits_nowaits_timeout    | 300 seconds | Specifies the number of seconds after which the System Waits By Number Of Waits pie chart is auto-refreshed in the System Waits Analysis dashboard.   |
| dash_sys_waits_timewait_timeout   | 300 seconds | Specifies the number of seconds after which the System Waits By Time Waited pie chart is auto-refreshed in the System Waits Analysis dashboard.   |
| dash_sys_waits_waitdtl_timeout    | 300 seconds | Specifies the number of seconds after which the System Waits Details table is auto-refreshed in the System Waits Analysis dashboard.  |
| deleted_charts_retention_time     | 7 days      | Specifies the number of days that a custom chart (displayed on a user-defined dashboard) is stored.   |
| deleted_probes_retention_time     | 7 days      | Specifies the number of days that a custom probe (displayed on a user-defined dashboard) is stored.   |
| download_chart_format             | jpeg        | Specifies the format in which a downloaded chart will be stored. May be jpeg or png.  |
| flapping_detection_state_change   | 3           | Specifies the number of state changes detected within a specified interval to define a given alert as flapping.- Flapping starts when more than <b>N</b> state changes have occurred over [ <b>N</b> + 1 * (min(probe_interval) * 2)] minutes and the fine state is not None. Where the default value of <b>N</b> is 2 or 3, and min(probe_interval) is the smallest interval for all the probes used by the alert.<br>- Flapping ends when ZERO state changes have occurred over [2 * <b>N</b> * min(probe_interval)] minutes. |
| job_retention_time                | 30 days     | Specifies the number of days that non-recurring scheduled tasks and their associated logs are retained, after their execution time.   |
| long_running_transaction_minutes  | 5 minutes   | Specifies the number of minutes a query executes for before being considered long running.  |
| nagios_cmd_file_name              | <file_name> | Specifies nagios command file to which passive service check result will be sent.   |
| nagios_enabled                    | t           | Specifies whether alert notification will be submitted to nagios or not.  |
| nagios_medium_alert_as_critical   | f           | Specifies whether medium level PEM alert will be considered as critical in nagios.  |
| nagios_spool_retention_time       | 7 days      | Specifies the number of days to retain nagios messages in the spool table before they are discarded.  |
| reminder_notification_interval    | 24 hours    | Specifies the number of hours after which a reminder email is sent in case an alert has not been cleared.   |
| server_log_retention_time         | 30 days     | Specifies the number of days that the server log is retained on the PEM server.   |
| show_data_tab_on_graph            | false       | If 'true', a Data tab is added to each graph. Select the Data tab to review the data that is plotted on the graph.  |
| show_objects_with_no_team         | true        | If 'true', and the team is not specified, then all the users can view the database servers, agents, and tools in the browser tree. Only the users with the 'pem_admin' role can view or change this parameter.  |

| Parameter name               | Value/Unit                              | Description   |
|------------------------------|---|---|
| smtp_authentication          | false                                   | Specifies whether to enable/disable authentication over SMTP.   |
| smtp_enabled                 | true                                    | Specifies whether to enable/disable sending of emails.  |
| smtp_encryption              | false                                   | Specifies whether to send SMTP email using an encrypted connection.   |
| smtp_password                |   | Specifies the password to be used to connect to the SMTP server.  |
| smtp_port                    | 25                                      | Specifies the SMTP server port to be used for sending email.  |
| smtp_server                  | 127.0.0.1                               | Specifies the SMTP server host address to be used for sending email.  |
| smtp_spool_retention_time    | 7 days                                  | Specifies the number of days to retain sent email messages in the spool table before they are discarded.  |
| smtp_username                |   | Specifies the username to be used to connect to SMTP server.  |
| smtp_message_linebreak       | LF                                      | Specifies the linebreak to be used in email message body.   |
| snmp_community               | public                                  | Specifies the SNMP community used when sending traps. Used only with SNMPv1 and SNMPv2.   |
| snmp_enabled                 | true                                    | Specifies whether to enable/disable sending SNMP traps.   |
| snmp_port                    | 162                                     | Specifies the SNMP server port to be used for sending SNMP traps.   |
| snmp_server                  | 127.0.0.1                               | Specifies the SNMP server host address to be used for sending SNMP traps.   |
| snmp_spool_retention_time    | 7 days                                  | Specifies the number of days to retain sent traps in the spool table before they are discarded.   |
| snmp_security_name           |   | Specifies the user name or security name for sending SNMP traps. Used only with SNMPv3.   |
| snmp_security_engine_id      |   | Specifies the Engine id of the SNMP Agent on the SNMP Server. Used only with SNMPv3.  |
| snmp_security_level          | NOAUTH_NOPRIV                           | Specifies Security level and its possible values can be: AUTH_NOPRIV - Authentication, No Privacy AUTH_PRIV - Authentication, Privacy NOAUTH_NOPRIV - no Authentication, no Privacy. Used only with SNMPv3. |
| snmp_context_name            |   | Specifies the Context name, the identifier for MIB objects when sending SNMP traps. Used only with SNMPv3.  |
| snmp_context_engine_id       |   | Specifies the Context engine id, the identifier for MIB objects when sending SNMP traps. If not specified, snmp_security_engine_id will be used. Used only with SNMPv3.                                     |
| snmp_authentication_protocol | NONE                                    | Specifies the authentication type for SNMP traps. Its possible values can be NONE, HMACMD5 or HMACSHA. Used only with SNMPv3.   |
| snmp_privacy_protocol        | NONE                                    | Specifies the privacy protocol for SNMP traps. Its possible values can be NONE, DES, AES128, IDEA, AES192, or AES256. Used only with SNMPv3.  |
| snmp_authentication_password |   | Specifies the authentication password associated with security name mentioned in snmp_security_name. Used only for SNMPv3.  |
| snmp_privacy_password        |   | Specifies the privacy password associated with security name mentioned in snmp_security_name. Used only for SNMPv3.   |
| webclient_help_pg            | EnterpriseDB<br>hosted<br>documentation | Specifies the location of the online PostgreSQL core documentation.   |

### 34.4.3 Log Manager

Use the Log Manager wizard to specify logging preferences for a Postgres database server. Log Manager supports Advanced Server and PostgreSQL versions 9.0 (and later). The Log Manager wizard assists in modifying configuration parameters that control:

- Where log files are written.
- How often log files are written.
- The type of information written to log files.
- The format of log file entries.

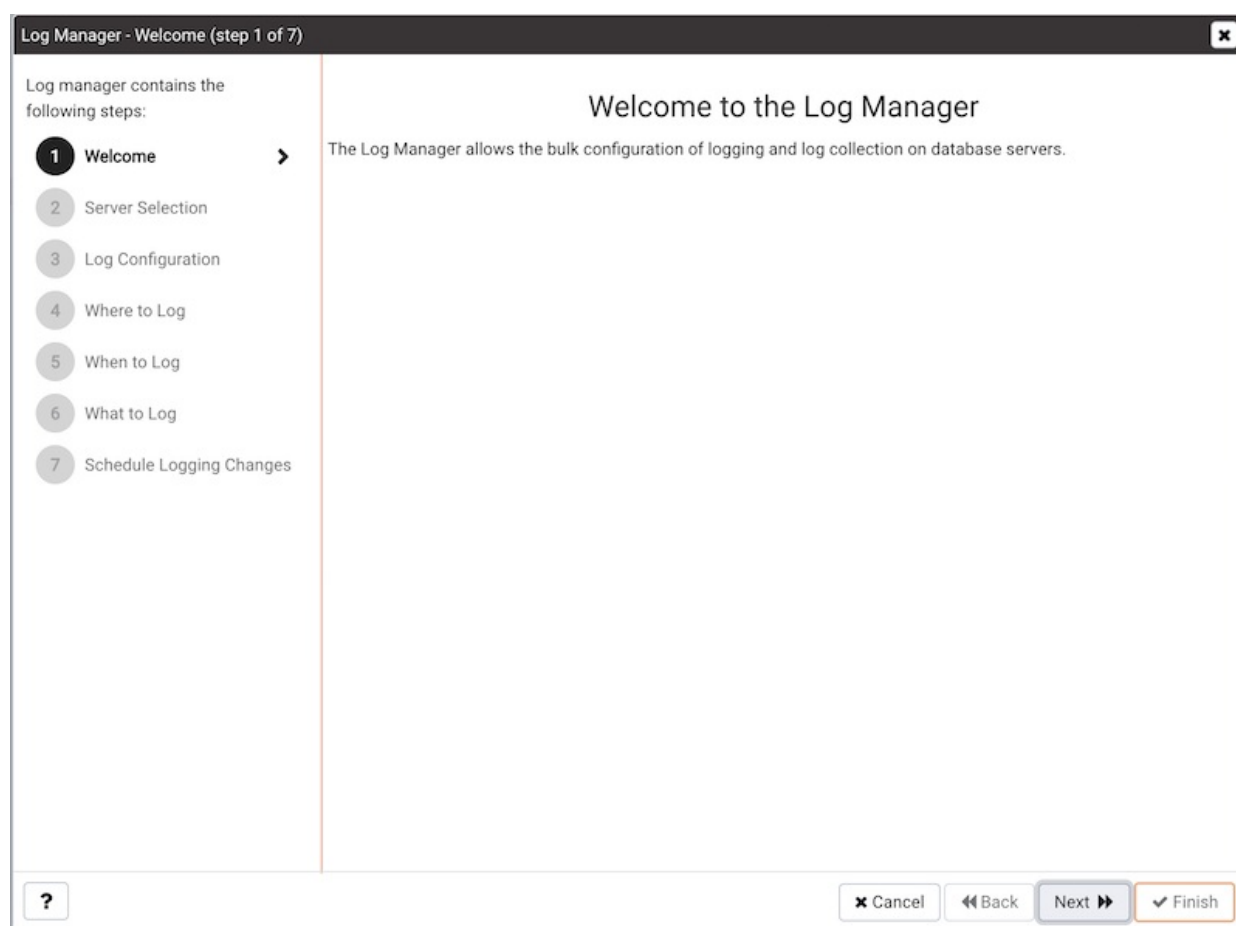
Before using Log Manager to define logging properties for a server, you must specify the name of the associated Advanced Server or PostgreSQL database server in the **Service ID** field on the **Advanced** tab of the **New Server Registration** (or **Properties**) dialog. If you do not specify the name of the service in the **Service ID** field, the server will not be made available for configuration on the **Server Selection** dialog.

For example, if you are setting logging preferences for an Advanced Server 9.4 instance that resides on a Linux host, set the **Service ID** field on the **Advanced** tab of the **Properties** dialog for the monitored server to **ppas-9.4**.

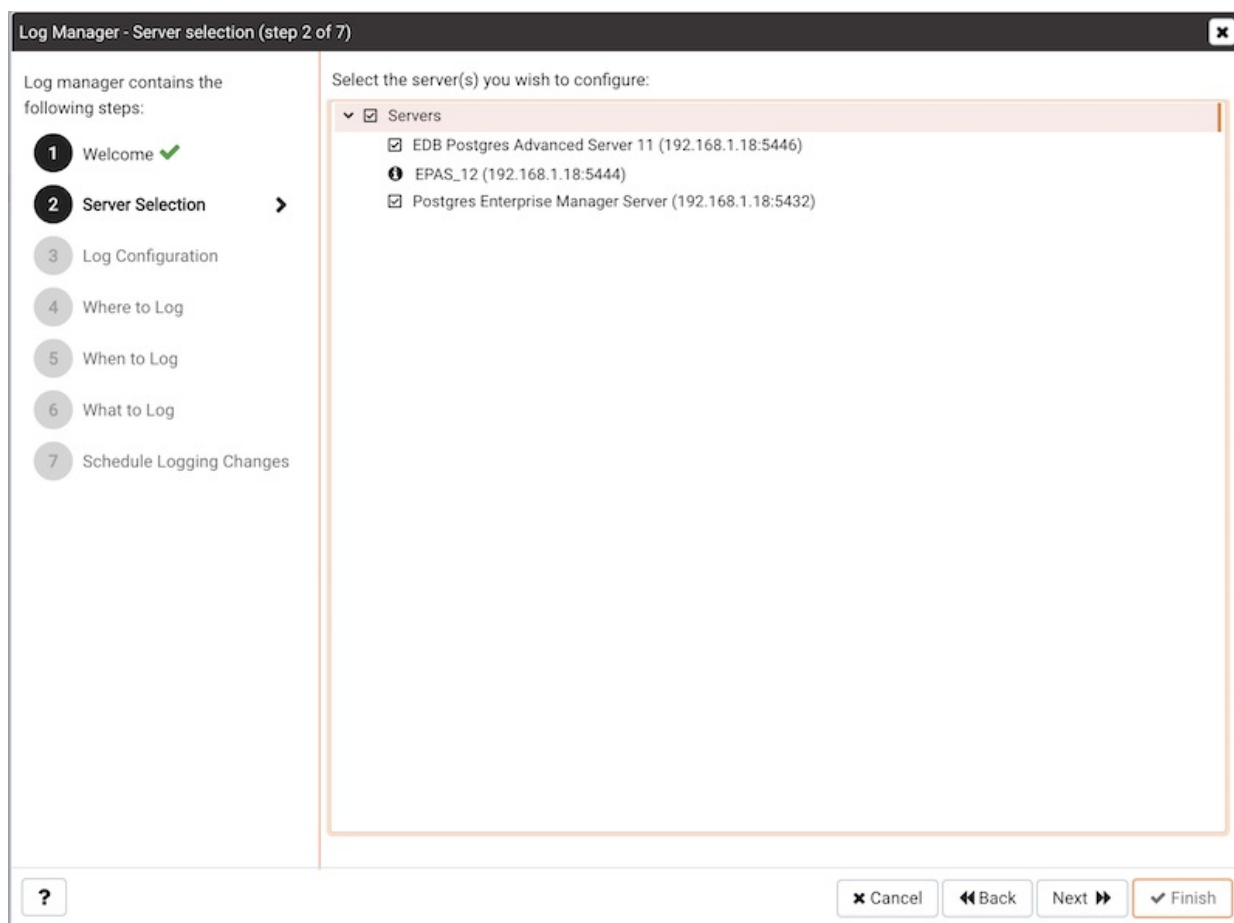
Note:

- Log manager depends on **Settings** & **Server log Configuration** probes to populate all the fields in the wizard. Therefore, ensure that those probes for selected servers are enabled. In addition, the execution frequency for those probes is set to a minimum to ensure that the log manager reflects the latest log configurations.
- Please re-run the log manager if you have done any changes related to logging in the configuration files (e.g. **postgresql.conf**) manually. PEM does not reflect those changes automatically.

To run the Log Manager, select the **Log Manager** menu option from the **Management** menu of the PEM client. The wizard opens, welcoming you to the Log Manager:



Click **Next** to continue to the **Server selection** dialog:



The **Server selection** dialog displays a list of the server connections monitored by PEM. Check the box next to the name of a server (or servers) to which the Log Manager wizard will apply the specified configuration. Log Manager is disabled for any server displaying a red exclamation mark to the left of its name in the Server selection tree control; there are several reasons that a server may not be enabled:

- Only a server that specifies a **Service ID** on the **Advanced** tab of the **Properties** dialog can be configured by Log Manager.

To provide a service ID, right click on the server name in the tree control, and select **Disconnect Server** from the context menu; if prompted, provide a password. Then, open the context menu for the server, and select **Properties**. Navigate to the **Advanced** tab, and provide the name of the service in the **Service ID** field; click **Save** to save your change and exit the dialog.

- If the PEM agent bound to the server does not have sufficient privileges to restart the server, the server will be disabled.
- If the PEM agent bound to the server is an older version than the associated PEM server, the server will be disabled.

Click **Next** to continue:



Log Manager - Log configuration (step 3 of 7)

Log manager contains the following steps:

- Welcome ✓
- Server Selection ✓
- Log Configuration** >
- Where to Log
- When to Log
- What to Log
- Schedule Logging Changes

**Import logs**

Import logs to PEM? ☐ No

Import frequency: 1 Hour

**Log rotation configuration**

Rotation size (MB): 10  
Automatic rotation of log files will happen after that much log output. 0 disables.

Rotation time (days): 1  
Automatic rotation of log files will happen after that time. 0 disables.

Truncate on rotation? ☐ No  
If on, an existing log file with the same name as the new log file will be truncated rather than appended to. Such truncation only occurs on time-driven rotation.

? Cancel Back Next Finish

Use options within the **Import logs** box to specify how often log files will be imported to PEM.

- Set the **Import logs to PEM** switch to **Yes** to specify that log files will be imported to PEM, and displayed on the Server Log Analysis dashboard.
- Use the **Import Frequency** drop-down list box to specify how often log files are imported to PEM. This option is only enabled when the **Import logs to PEM** option is enabled. The default value is 5 minutes.

Use the options in the **Log rotation configuration** box to specify the maximum length (lifespan or size) of a log file.

- Use the **Rotation size** field to specify the maximum size in megabytes of an individual log file. The default value is 10 MB; when set to 0, no limit is placed on the maximum size of a log file.
- Use the **Rotation time** field to specify the number of whole days that should be stored in each log file. The default value is 1 day.

Use the **Truncate on Rotation** switch to specify server behavior for time-based log file rotation:

- Select **ON** to specify that the server should overwrite any existing log file that has the same name that a new file would take.
- Select **OFF** to specify that the server should append any new log file entries to an existing log file with the same name that a new log file would take. This is the default behavior.

Click **Next** to continue to the **Where to Log** dialog:



Use the fields on the **Where to Log** dialog to specify where log files should be written. Select an option from the **Log destination** box to specify a destination for the server log output:

- Set the **stderr** switch to **Yes** to specify that log files should be written to **stderr**. By default, server log entries are written to **stderr**.
- Set the **csvlog** switch to **Yes** to specify that log files should be written to file in a comma-separated value format. This option is automatically enabled (and no longer editable) if you have selected **Import logs to PEM** on the **Schedule** dialog; if you are not importing server log files to PEM, this option is editable.
- Set the **syslog** switch to **Yes** to specify that log files should be written to the system log files.
- On Windows, set the **eventlog** switch to **Yes** to specify that log files should be written to the event log.

Use options in the **Log collection** box to specify collection preferences. Use the **Log Collector** switch to instruct the server to re-direct captured log messages (directed to STDERR) into log files:

- Specify **Enable** to instruct the server to re-direct captured error messages to a log file. By default, Log Collector is enabled.
- Specify **Disable** to instruct the server that it should not re-direct error messages to a log file.

Use the **Log Silent Mode** switch to instruct the server to run silently in the background, disassociated from the controlling terminal:

- Select **Enable** to instruct the server to run silently in the background.
- Select **Disable** to instruct the server to display log file entries on the controlling terminal as each log entry is written.

Use options in the Log Directory box to specify log file location preferences:

- Set the **Change log directory for selected servers?** switch to **Yes** to specify that each set of log files should be maintained in a separate directory.
- When **Log Collector** is enabled, you can use the **Directory name** field to specify the directory to which the log file will be written. By default, logs are written to the **pg\_log** directory under the installation directory of the monitored server.

When **Import logs to PEM** is disabled, you can use the **Log file name** field to specify the filename to which the logs will be written. The

'DEFAULT' value in the **Log File Name** field represents 'postgresql-%Y-%m-%d\_%H%M%S.log' for all the PostgreSQL servers and 'enterprisedb-%Y-%m-%d\_%H%M%S.log' for all the Postgres Plus Advanced Servers.

When logging to **syslog** is enabled, you can use the **Syslog facility** drop-down list box to specify which syslog facility should be used.

When logging to **syslog** is enabled, you can use the **Syslog ident** field to specify the program name that will identify Advanced Server entries in system logs.

Note: Changing **eventlog** & **Log Collector** parameter requires database server restart

Click **Next** to continue:

**Log Manager - When to log (step 5 of 7)**

Log manager contains the following steps:

- Welcome ✓
- Server Selection ✓
- Log Configuration ✓
- Where to Log ✓
- When to Log** >
- What to Log
- Schedule Logging Changes

|                                  |         |   |
|----------------------------------|---------|---|
| Client min messages              | notice  | Message levels sent to the client.  |
| Log min messages                 | warning | Message levels written to the server log.   |
| Log min error statement          | error   | SQL statements that cause an error condition are recorded in the server log.                                |
| Log min duration statement (ms)  | -1      | Logs only statements running at least this number of milliseconds. -1: disabled, 0 logs all with durations. |
| Log temp files (kilobytes)       | -1      | Log temporary files equal or larger than the specified size; -1 disables, 0 logs all temp files.            |
| Log AutoVacuum min duration (ms) | -1      | If auto-vacuuming exceeds the length of time specified, the activity will be logged. The default            |

? Cancel Back Next Finish

Use the fields on the **When to Log** dialog to specify which events will initiate a log file entry. The severity levels (in order of severity, from most severe to least severe) are:

| Severity              | Description  |
|-----------------------|--|
| panic                 | Errors that cause all database sessions to abort.                                  |
| fatal                 | Errors that cause a session to abort.  |
| log                   | Information messages of interest to administrators.                                |
| error                 | Errors that cause a command to abort.  |
| warning               | Error conditions in which a command will complete but may not perform as expected. |
| notice                | Items of interest to users. This is the default.                                   |
| info                  | Information implicitly requested by the user.                                      |
| debug5 through debug1 | Detailed debugging information useful to developers                                |

- Use the **Client min messages** drop-down list box to specify the minimum severity level that will be sent to the client application. The default value is **notice**.
- Use the **Log min messages** drop-down list box to specify the minimum severity level that will be written to the server log. The default value is **warning**.
- By default, when an error message is written to the server log, the text of the SQL statement that initiated the log entry is not included. Use the **Log min error statement** drop-down list box to specify a severity level that will trigger SQL statement logging. If a message is of the specified severity or higher, the SQL statement that produced the message will be written to the server log. The default value is **error**.
- Use the **Log min duration statement** field to specify a statement duration (in milliseconds); any statements that exceed the specified number of milliseconds will be written to the server log. The length of time that it took for the statement to execute will be included in the log entry. A value of -1 disables all duration-based logging; a value of 0 logs all statements and their duration. The default value is **-1**.
- Use the **Log temp files** field to specify a file size in kilobytes; when a temporary file reaches the specified size, it will be logged. The default value is **-1**.
- Use the **Log autoVacuum min duration** field to specify a time length in milliseconds; if auto-vacuuming exceeds the length of time specified, the activity will be logged. The default value is **-1**.

Click **Next** to continue to the **What to log** dialog:

Log Manager - What to log (step 6 of 7)

Log manager contains the following steps:

- Welcome ✓
- Server Selection ✓
- Log Configuration ✓
- Where to Log ✓
- When to Log ✓
- What to Log** >
- Schedule Logging Changes

**Debug options**

Parse tree?       Rewriter output?

Execution plan?

Indent debug options output in log?

**General options**

Checkpoints?       Connections?

Disconnections?       Duration?

Hostname?       Lock waits?

Error verbosity:   
Level of message detail written to the server log.

Prefix string:   
Use the Prefix String field to specify a printf-style string that is written at the beginning of each log file entry.

Statements:   
Controls which SQL statements are logged.

?                       

Use the fields on the **What to log** dialog to specify log entry options that are useful for debugging and auditing.

The switches in the **Debug options** box instruct the server to include information in the log files related to query execution that may be of interest to a developer:

- Set the **Parse tree** switch to **Yes** to instruct the server to include the parse tree in the log file. The default value is **No**.
- Set the **Rewriter output** switch to **Yes** to instruct the server to include query rewriter output in the log file. The default value is **No**.
- Set the **Execution plan** switch to **Yes** to instruct the server to include the execution plan for each executed query in the log file. The default value is **No**.


By default, **Indent Debug Options Output in Log** option is set to **No**. When this option is enabled, the server indents each line that contains a parse tree entry, a query rewriter entry or query execution plan entry. While indentation makes the resulting log file more readable, it results in a longer log file. To enable indentation of log file entries related to debugging, move the switch to **Yes**.

Use the switches in the **General options** box to instruct the server to include auditing information in the log file:

- Set the **Checkpoints** switch to **Yes** to include checkpoints and restartpoints in the server log. By default, this is set to **No**.
- Set the **Connections** switch to **Yes** to include each attempted connection to the server (as well as successfully authenticated connections) in the server log. By default, this is set to **No**.
- Set the **Disconnections** switch to **Yes** to include a server log entry for each terminated session that provides the session information and session duration. By default, this is set to **No**.
- Set the **Duration** switch to **Yes** to include the amount of time required to execute each logged statement in the server log. By default, this is set to **No**.
- Set the **Hostname** switch to **Yes** to include both the IP address and host name in each server log entry (by default, only the IP address is logged). Please note that this may cause a performance penalty. By default, this is set to **No**.
- Set the **Lock Waits** switch to **Yes** to instruct the server to write a log entry for any session that waits longer than the time specified in the **deadlock\_timeout** parameter to acquire a lock. This is useful when trying to determine if lock waits are the cause of poor performance. By default, this is set to **No**.
- Use the **Error verbosity** drop-down list box to specify the detail written to each entry in the server log.
  - Select **default** to include the error message, DETAIL, HINT, QUERY and CONTEXT in each server log entry.
  - Select **terse** to log only the error message, excluding the DETAIL, HINT, QUERY and CONTEXT information from each server log entry.
  - Select **verbose** to include the error message, the DETAIL, HINT, QUERY and CONTEXT error information, SQLSTATE error code and source code file name, the function name, and the line number that generated the error.
- Use the **Prefix string** field to specify a printf-style string that is written at the beginning of each log file entry. The **Escape** characters in the following table represent the information described in the **Information** column. Some information is available to **Session** processes only; **Helper** processes can provide all of the information specified in the **Prefix String**. The default value is %t (timestamp without milliseconds).

You can include:

| Escape | Information                                     | Session/Helper |
|--------|---|----------------|
| %a     | Application Name                                | Session        |
| %u     | User Name                                       | Session        |
| %d     | Database Name                                   | Session        |
| %r     | Remote host name or IP address, and remote port | Session        |

|    |   |         |
|----|---|---------|
| %h | Remote host name or IP address  | Session |
| %p | Process ID  | Helper  |
| %t | Time stamp without milliseconds   | Helper  |
| %m | Time stamp with milliseconds  | Helper  |
| %i | Command tag: type of statement that generated the log entry   | Session |
| %e | SQLSTATE error code   | Helper  |
| %c | Session identifier  | Helper  |
| %l | Line number of the log entry  | Helper  |
| %s | Process start time stamp  | Helper  |
| %v | Virtual transaction ID (backendID/localXID)   | Helper  |
| %x | Transaction ID (  if not assigned)             | Helper  |
| %q | Produces no output, but instructs non-session processes to stop at this point in the string; will be ignored by session processes | Helper  |
| %% | Literal %   | Helper  |

- Use the **Statements** drop-down list box to specify which SQL statements will be included in the server log. The default is **none** ; valid options

are:

- **none** - Specify **none** to disable logging of SQL statements.
- **ddl** - Specify **ddl** to instruct the server to log ddl (data definition language) statements, such as CREATE, ALTER, and DROP.
- **mod** - Specify **mod** to instruct the server to log all **ddl** statements, as well as all **dml** (data modification language) statements, such as INSERT, UPDATE, DELETE, TRUNCATE and COPY FROM.
- **all** - Specify **all** to instruct the server to log all SQL statements.

Click **Next** to continue:

Log Manager - Schedule logging changes (step 7 of 7)

Log manager contains the following steps:

- Welcome ✓
- Server Selection ✓
- Log Configuration ✓
- Where to Log ✓
- When to Log ✓
- What to Log ✓
- Schedule Logging Changes >

Configure logging now? **Yes**

Time

Note: The selected servers will be restarted to complete log configuration.

? Cancel Back Next Finish

Use the options on the **Schedule Logging Changes** dialog to select a time that logging configuration changes will be applied. Note that when you apply the configuration changes specified with the Log Manager wizard, the server will be restarted, temporarily interrupting use of the database server for users.

- Set the **Configure Logging Now** switch to **Yes** to specify that PEM will configure logging and restart the server when you have completed the Log Manager wizard.
- Set the **Configure Logging Now** switch to **No** and use the **Schedule it for some other time** date selector to specify a convenient time for the server to restart.

Click **Finish** to complete the wizard, and either restart the server, or schedule the server restart for the time specified on the scheduling dialog.

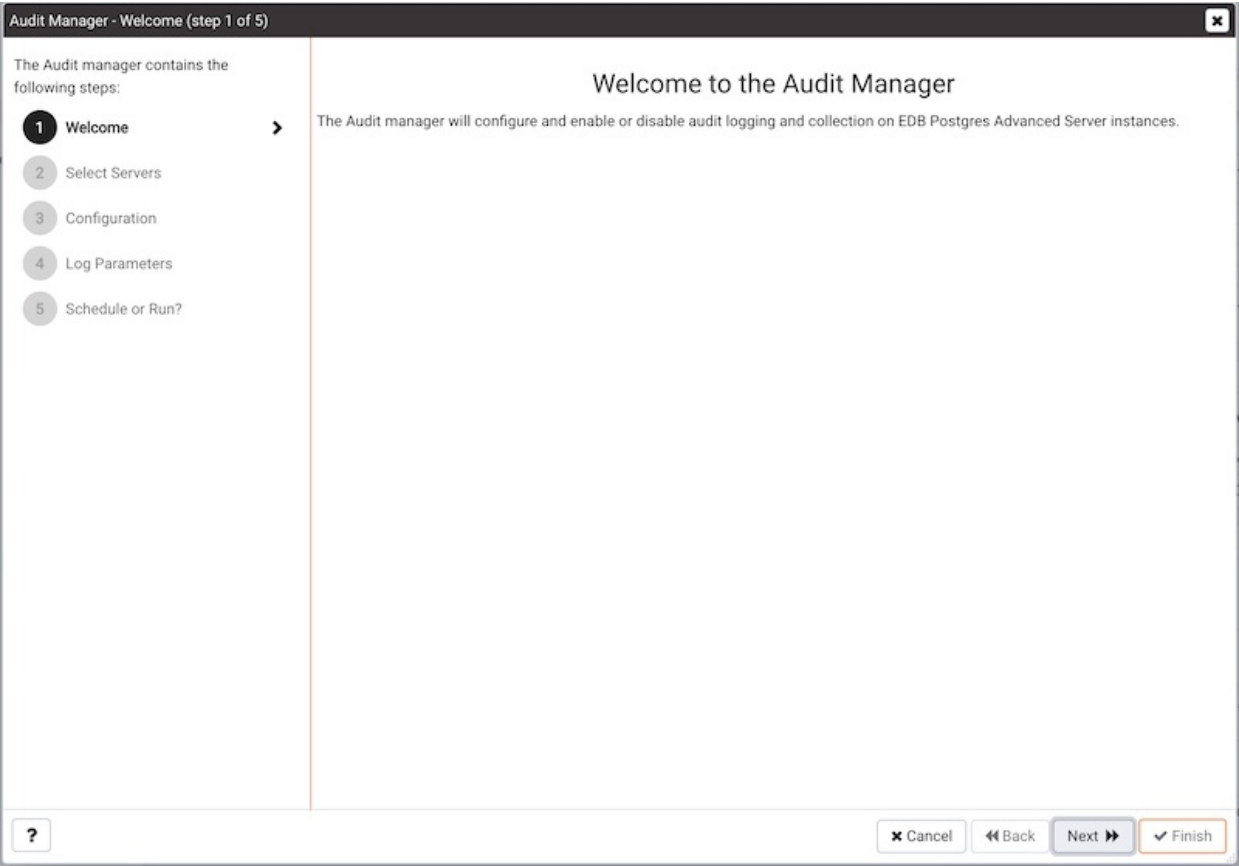
When you have completed the Log Manager wizard, you can use the **Scheduled Tasks** dialog to confirm that the configuration file update and server restart have been scheduled.

### 34.4.4 Audit Manager

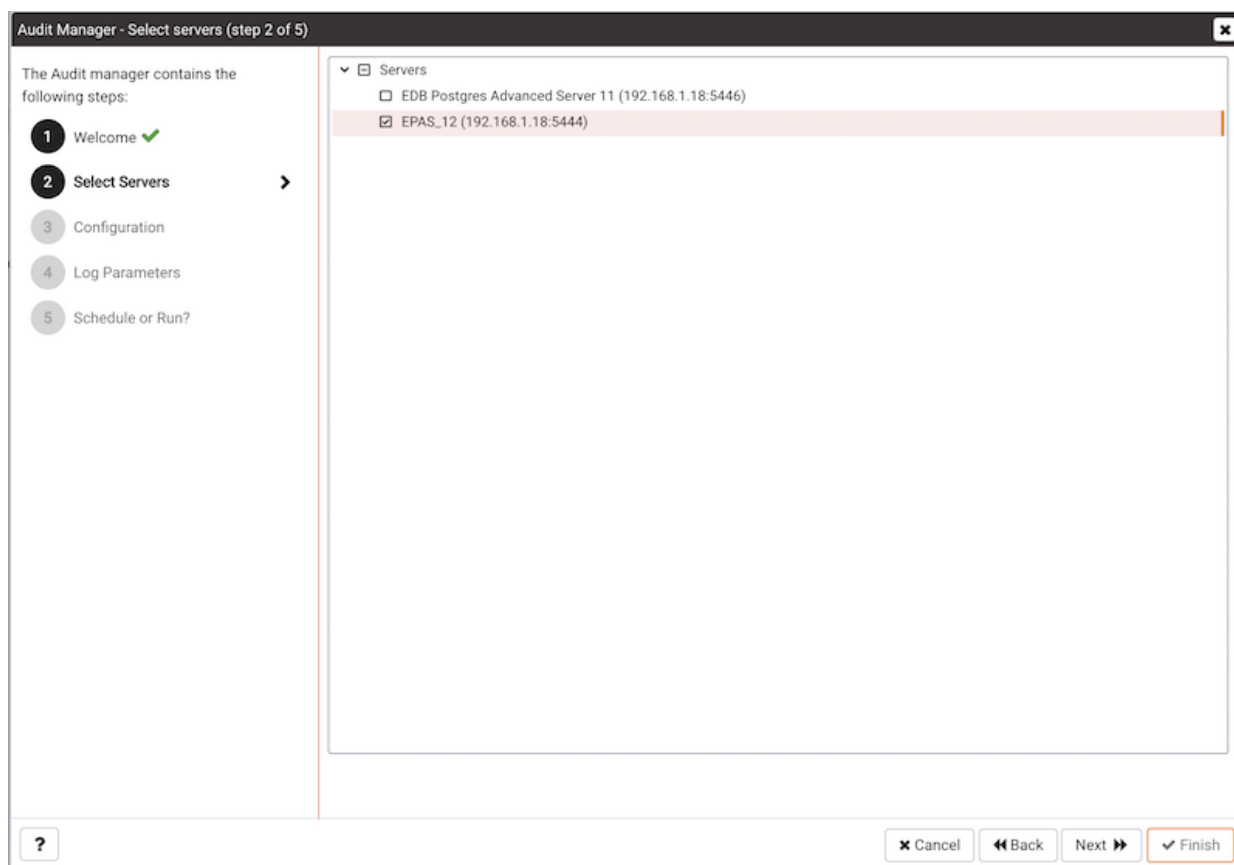
You can use the PEM Audit manager to configure, enable, and disable audit logging of EDB Postgres Advanced Server instances. The Audit manager also

enables audit log collection, allowing you to view log data on the [Audit Log Dashboard](#).

To run the Audit manager wizard, select **Audit manager...** from the PEM client **Management** menu. Audit manager opens, displaying the **Welcome** dialog:

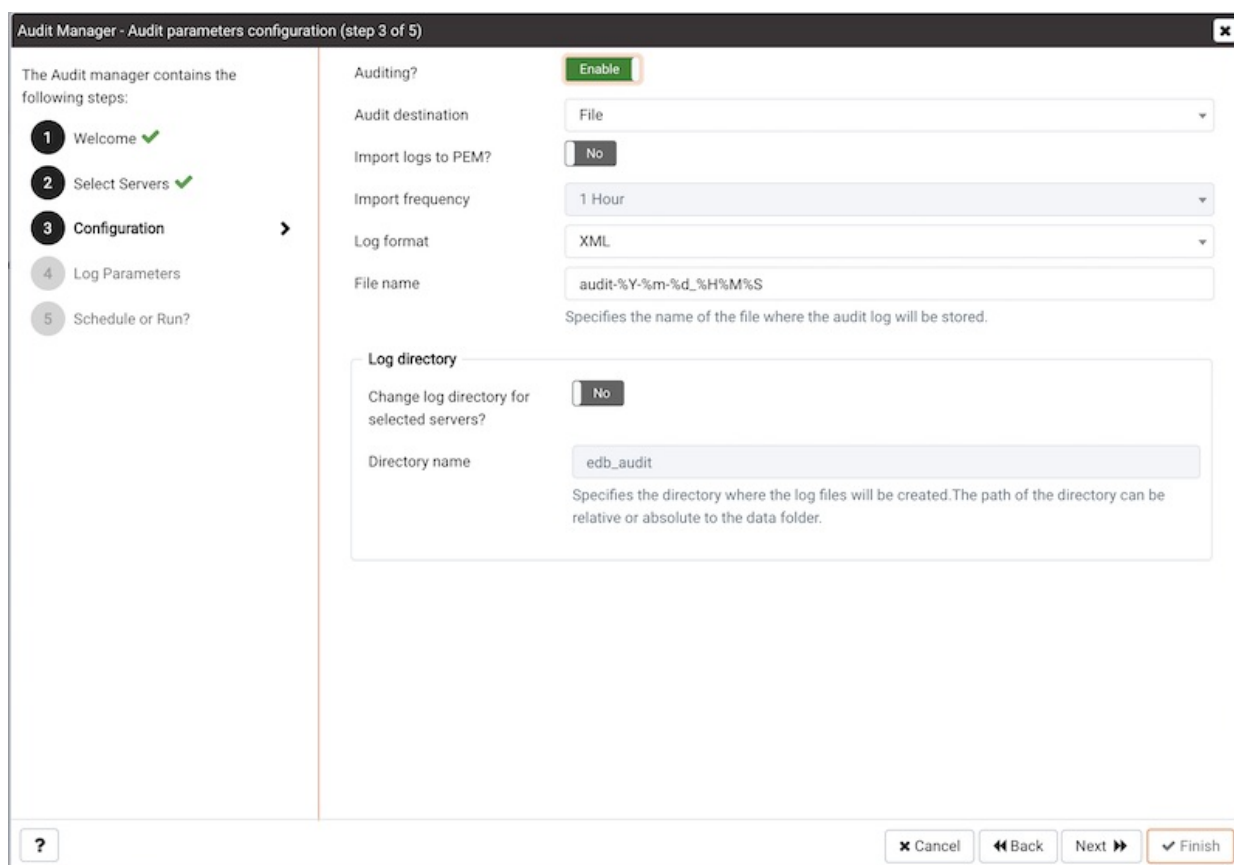


Click **Next** to continue:



Use the **Select servers** tree control to specify the servers to which the auditing configuration will be applied. To make a server available in the tree control, you must provide the **Service ID** on the PEM **Server** dialog. Note that only EDB Postgres Advanced Server supports auditing; PostgreSQL servers will not be included in the tree control.

Click **Next** to continue:





Use the controls on the **Audit parameters configuration** dialog to specify configuration details that will be applied to each server:

- Use the **Auditing** switch to **Enable** or **Disable** auditing on the specified servers.
- Use the **Audit destination** drop-down to select a destination for the audit logs; select **File** or **Syslog**. Please note this feature is supported on Advanced Server 10 and newer releases only.
- Use the **Import logs to PEM** switch to instruct PEM to periodically import log records from each server to the PEM Server. Set the switch to **Yes** to import log files; the default is **No**.
- Use the **Import frequency** drop-down listbox to specify how often PEM will collect log records from monitored servers when log collection is enabled.
- Use the **Log format** drop-down listbox to select the raw log format that will be written on each server. If log collection is enabled, the PEM server will use CSV format.
- Use the **File name** field to specify the format used when generating log file names. By default, the format is set to **audit-%Y-%m-%d\_%H%M%S** if log collection is enabled.

Use fields in the **Log directory** box to specify information about the directory in which the log files will be saved:

- Move the **Change log directory for selected servers?** switch to **Yes** to enable the **Directory name** field.
- Use the **Directory name** field to specify the name of the directory on each server into which audit logs will be written. The directory specified will be created as a sub-directory of the **data** directory on the server.

Click **Next** to continue:

**Audit Manager - Audit log configuration (step 4 of 5)**

The Audit manager contains the following steps:

- 1 Welcome ✓
- 2 Select Servers ✓
- 3 Configuration ✓
- 4 Log Parameters >
- 5 Schedule or Run?

**Connection attempts**

☒ None ☐ All ☐ Failed

Enables auditing of database connection attempts by users. To disable auditing of all connection attempts, set Connection attempts to none. To audit all failed connection attempts, set the value to failed.

**Disconnection attempts**

☒ None ☐ All

Enables auditing of database disconnections by connected users. To enable auditing of disconnections, set the value to all. To disable, set the value to none.

**Log statements**

☒ Select ☒ Error ☒ DML ☒ DDL

This configuration parameter is used to specify auditing of different categories of SQL statements.

**Audit tag**

Use this configuration parameter to specify a string value that will be included in audit log when the edb\_audit parameter is set to CSV or XML.

**Log rotation**

Enable? ☐ No

Day

Size(MB)

Time(seconds)

**Buttons:** [?] [Cancel] [Back] [Next] [Finish]

The **Audit log configuration** dialog is only available if you have specified a value of **Enable** in the **Auditing** field. Use the controls on the **Audit log configuration** dialog to specify log configuration details that will be applied to each server:

- Use the **Connection attempts** switch to specify if connection attempts should be logged. Specify: **None** to disable connection logging, **All** to indicate that all connection attempts will be logged, or **Failed** to log any connection attempts that fail.
- Use the **Disconnection attempts** switch to specify if disconnections should be logged. Specify **None** to specify that disconnections should not be logged, or **All** to enable disconnection logging.

- Use the **Log statements** field to specify the statement types that will be logged. Click within the field, and select from:
  - Select - All statements that include the SELECT keyword will be logged
  - Error - All statements that result in an error will be logged.
  - DML - All DML (Data Modification Language) SQL statements will be logged.
  - DDL - All DDL (Data Definition Language) SQL statements (those that add, delete or alter data) will be logged.
  - Check the box next to **Select All** to select all statement types.
  - Check the box next to **Unselect All** to deselect all statement types.
- Use the **Audit tag** field to specify a tracking tag for the collected logs. Please note that audit tagging functionality is available only for Advanced Server versions 9.5 and later. If you are defining auditing functionality for multiple servers, and one or more of the servers are version 9.5 or later, this field will be enabled, but if selected, tagging functionality will only apply to those servers that are version 9.5 or later.

Use the fields in the **Log rotation** box to specify how the log files are managed on each server:

- Use the **Enable?** switch to specify that logfiles should be rotated. Please note that a new log file should be used periodically to prevent a single file becoming unmanageably large.
- Use the **Day** drop-down listbox to select a day or days on which the log file will be rotated.
- Use the **Size (MB)** field to specify a size in megabytes at which the log file will be rotated.
- Use the **Time (seconds)** field to specify the number of seconds between log file rotations.

Click **Next** to continue:

Audit Manager - Schedule auditing changes (step 5 of 5)

The Audit manager contains the following steps:

- 1 Welcome ✓
- 2 Select Servers ✓
- 3 Configuration ✓
- 4 Log Parameters ✓
- 5 Schedule or Run? >

Configure logging now? **Yes**

Time?

NOTE: The selected servers will be restarted to complete audit configuration.

? Cancel Back Next Finish

Use the **Schedule auditing changes** dialog to specify when the new configuration will be applied to the servers:

- Set the **Configure logging now?** switch to **Yes** to apply the configuration immediately.
- Use the **Time?** selector to schedule the audit configuration for a later time; use the date and time selectors to specify the date and time at which the PEM server will apply the configuration.

Click the **Finish** button to schedule a job to apply the configuration to each server. The job will consist of two tasks. One task will update the audit logging configuration on the server, and one task will reload the server with the new configuration.

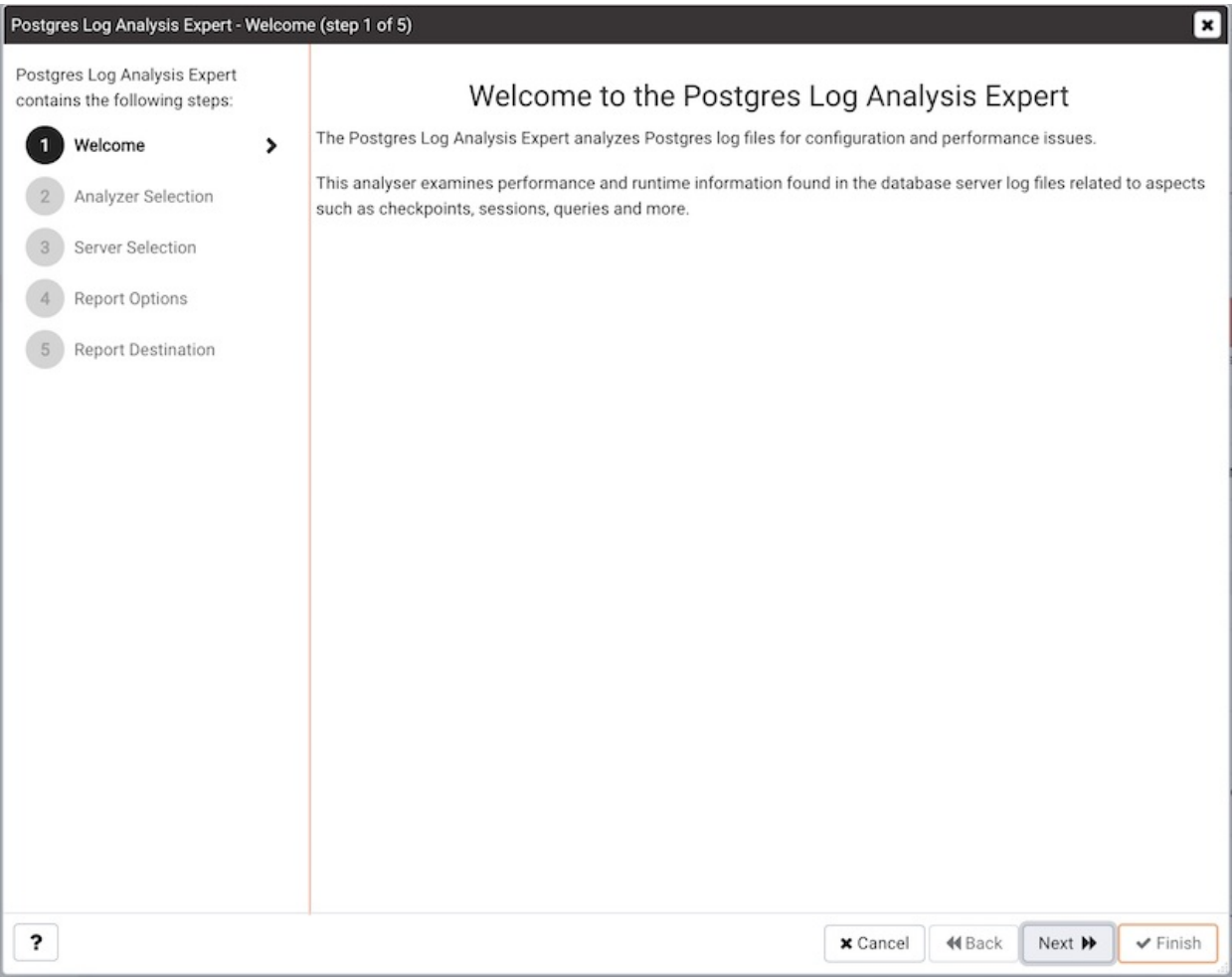
The scheduled jobs can be viewed in theTask Viewer, and the results in theLog Viewer when opened from the appropriate server or agent.

### 34.4.5 Postgres Log Analysis Expert

The Postgres Log Analysis Expert analyzes the log files of servers that are registered with PEM, and produces a report that provides an overview of your Postgres cluster’s usage based on log file entries. You can use information on the Log Analysis Expert reports to make decisions about optimizing your cluster usage and configuration to improve performance.

Before invoking the Postgres Log Analysis Expert, you must specify the Service ID on the Advanced tab of the server’s properties dialog, and use the Log Manager wizard to enable log collection by the PEM server. To invoke the Log Manager wizard, select the Log Manager... option from the Management menu; check the box next to Import logs to PEM in the Import Logs panel of the wizard to enable log collection.

To open the Postgres Log Analysis Expert wizard, select the Postgres Log Analysis Expert... option from the Management menu of the PEM client. When the wizard’s Welcome dialog opens, click Next to continue.



The wizard’s Analyzer selection dialog displays a list of Analyzers from which you can select. Each Analyzer generates a corresponding table, chart, or graph that contains information gleaned from the log files.

Postgres Log Analysis Expert - Analyzer Selection (step 2 of 5)

Postgres Log Analysis Expert contains the following steps:

- 1 Welcome ✓
- 2 Analyzer Selection ➤
- 3 Server Selection
- 4 Report Options
- 5 Report Destination

Select the analyzers or individual analyzer you wish to run:

- ☒ Analyzers
  - ☒ Autoanalyze Statistics
  - ☒ Autovacuum Statistics
  - ☒ CHECKPOINT Statistics
  - ☒ COMMIT/ROLLBACK Statistics
  - ☒ Connections Overview
  - ☒ DDL Statistics
  - ☒ DML Statistics
  - ☒ Frequently Executed Query Statistics
  - ☒ Hourly DML Statistics
  - ☒ Lock Statistics
  - ☒ Log Event Statistics
  - ☒ Log Statistics
  - ☒ Most Time Consumed Query Statistics
  - ☒ Slow Running Query Statistics
  - ☒ Summary Statistics
  - ☒ Temporary File Statistics
  - ☒ Temporary Query Statistics
  - ☒ Waiting Statistics

? ✕ Cancel ⏪ Back Next ⏩ ✓ Finish

Check the box to the left of an Analyzer to indicate that the Log Analysis Expert should prepare the corresponding table, chart or graph. After making your selections, click **Next** to continue to the **Server selection** tree control.

Postgres Log Analysis Expert - Server Selection (step 3 of 5)

Postgres Log Analysis Expert contains the following steps:

- 1 Welcome ✓
- 2 Analyzer Selection ✓
- 3 Server Selection >
- 4 Report Options
- 5 Report Destination

Select the servers you wish to analyze:

✓ Servers

- ❗ EDB Postgres Advanced Server 11(192.168.1.18:5446)
- ❗ EPAS\_12(192.168.1.18:5444)
- ❗ PGSQL12\_Centos7\_1(192.168.1.12:5432)
- ☑ Postgres Enterprise Manager Server(192.168.1.18:5432)

? Cancel Back Next Finish

Use the tree control to specify which servers you would like the Postgres Log Analysis Expert to analyze. If you select multiple servers, the resulting report will contain the corresponding result set for each server in a separate (but continuous) list. Click [Next](#) to continue.

Postgres Log Analysis Expert - Report Options (step 4 of 5)

Postgres Log Analysis Expert contains the following steps:

- Welcome ✓
- Analyzer Selection ✓
- Server Selection ✓
- Report Options** >
- Report Destination

### Time intervals

Relative days:

From:

To:

(+/-)From date:  Days before or after the 'From date' that should be included in the analysis.

### Options

Aggregate method:  Method to consolidate data for the selected time span.

Time span:  Number of minutes that the analyzer will incorporate into each calculation for a point on a graph.

Rows limit:  Number of rows to include in a table.

?

Use the fields in the **Options** section to specify the analysis method and the maximum length of any resulting tables:

- Use the **Aggregate method** drop-down to select the method used by the Log Analysis Expert to consolidate data for the selected time span - select from:
  - SUM** - **SUM** instructs the analyzer to calculate a value that is the sum of the collected values for the specified time span.
  - AVG** - **AVG** instructs the analyzer to calculate a value that is the average of the collected values for the specified time span.
  - MAX** - **MAX** instructs the analyzer to use the maximum value that occurs within a specified time span.
  - MIN** - **MIN** instructs the analyzer to use the minimum value that occurs within a specified time span.
- Use the **Time span** field to specify the number of minutes that the analyzer will incorporate into each calculation for a point on a graph. For example, if the **Time span** is '5 minutes', and the **Aggregate method** is 'AVG', each point on the given graph will contain the average value of the activity that occurred within a five minute time span.
- Use the **Rows limit** field to specify the maximum number of rows to include in a table.

Use the fields in the **Time Intervals** section to specify the time range that the Log Analysis Expert will analyze:

- Set **Relative days** to **Yes** to enable the **(+/-)From date** field and specify the number of days before or after the date and time selected in the **From** field.
- Use the **From** field to specify the starting date and time for the analysis.
- Use the **To** field to specify the ending date and time for the analysis.
- Use the **(+/-) From date** selector to specify the number of days before or after the **From** date that should be included in the analysis.

When you've specified the report options, click **Next** to continue.

Postgres Log Analysis Expert - Report Destination (step 5 of 5)

Postgres Log Analysis Expert contains the following steps:

- 1 Welcome ✓
- 2 Analyzer Selection ✓
- 3 Server Selection ✓
- 4 Report Options ✓
- 5 Report Destination >

☒ View the report now  
 Display the report on a new dashboard panel

☐ Download the report  
 Download the report as an HTML file

? Cancel Back Next Finish

You can select the default option ( **Finish** ) to view the Log Analysis Expert report in the PEM client's tabbed browser, or click the radio button next to **Download the report** to save a copy of the report to an HTML file for later use.

## Reviewing the Postgres Log Analysis Expert Report

If you've elected to review the report immediately, the Postgres Log Analysis Expert report will be displayed in the PEM Client window. If the report contains an analysis of more than one monitored server, the graphs will be displayed in sets; first the graphs, tables and charts that display statistics for one server, then the graphs for the next server in the report.

The Postgres Log Analysis Expert Report header displays the date and time that the report was generated, the time period that the report spans, and the Aggregation method specified when defining the report. The name of the server for which information is displayed is noted at the start of each section of the report.

The report displays the tables, graphs and charts that were selected in the Log Analysis Expert wizard. Use the **Jump To** button (located in the lower-right hand corner of the screen) to navigate to a specific graphic.

Postgres Log Analysis Expert

Interval: 2020-4-20 15:55:19 - 2020-4-27 15:55:19

Generated: 2020-04-27 16:00:51

Span: 5 Minutes

Aggregate: SUM

Go to: Postgres Enterprise Manager Server

▼ Postgres Enterprise Manager Server(192.168.1.18:5432)

Summary Statistics

| Settings                    | Values                               |
|-----------------------------|--------------------------------------|
| Number of unique queries    | 151649                               |
| Total queries               | 155045                               |
| Total queries duration      |                                      |
| First query                 | 27/04/2020 15:47:09.86 IST           |
| Last query                  | 27/04/2020 15:53:59.611 IST          |
| Queries peak time           | 27/04/2020 15:49:38 IST queries 2821 |
| Number of events            | 155045                               |
| Number of unique events     | 1                                    |
| Total number of sessions    | 348                                  |
| Total duration of sessions  |                                      |
| Average sessions duration   |                                      |
| Total number of connections | 0                                    |
| Total number of databases   | 0                                    |

Hourly DML Statistics

| Time             | Database name | Statement | Count | Min duration | Max duration | Avg duration |
|------------------|---------------|-----------|-------|--------------|--------------|--------------|
| 27/04/2020 15:00 | db01          | SELECT    | 80    | 0.05         | 105.40       | 4.90         |
| 27/04/2020 15:00 | edbstore_temp | SELECT    | 58    | 0.02         | 66.58        | 4.64         |
| 27/04/2020 15:00 | hr            | SELECT    | 48    | 0.01         | 29.26        | 1.87         |
| 27/04/2020 15:00 | pem           | COPY      | 1641  | 0.65         | 43.81        | 2.08         |
| 27/04/2020 15:00 | pem           | DELETE    | 73    | 0.44         | 8.74         | 1.04         |
| 27/04/2020 15:00 | pem           | INSERT    | 190   | 0.06         | 9.50         | 2.19         |

The report may include one or more of the following:

- The **Summary Statistics** table displays a summary of server activity for the selected server.
  - The **Number of unique queries** row displays the count of unique queries made against the selected server in the specified time period.
  - The **Total queries** row displays the count of queries made against the selected server in the specified time period.
  - The **Total queries duration** row displays the amount of time used to execute queries against the server.
  - The **First query** row displays the time (within the specified time period) that the first query executed against the server.
  - The **Last query** row displays the time (within the specified time period) that the last query executed against the server.
  - The **Queries peak time** row displays the point in time (within the specified time period) that query activity reached it's highest level.
  - The **Number of events** row displays the count of log events within the specified time period.
  - The **Number of unique events** row displays the count of unique server events.
  - The **Total number of sessions** row displays a count of the number of sessions recorded within the time period.
  - The **Total duration of sessions** row displays the amount of time that sessions were connected (during the specified time period).
  - The **Average sessions duration** row displays the average length of each session.
  - The **Total number of connections** row displays the number of user connections made to the server.
  - The **Total number of databases** row displays the number of databases on the selected server.
- The **Hourly DML Statistics** table displays the statistics related to the use of various DML commands (SELECT, INSERT, UPDATE, DELETE, COPY and FETCH) within a one-hour period. To generate values in the **Min Duration(sec)**, **Max Duration(sec)**, and **Avg Duration(sec)** columns of this table, you must specify a value greater than or equal to 0 in the log\_min\_duration\_statement configuration parameter. You can set the parameter by either modifying the **postgresql.conf** file with your editor of choice, or by specifying a value of 0 or greater in the **Log Min Duration Statement** field of the **Log Manager** wizard.
  - The **Time** column displays the start of the one-hour period for which data was analyzed.
  - The **Database** column displays the name of the database in which the specified DML command executed.
  - The **Command Type** column displays the DML command type.
  - The **Total Count** column displays the number of times that a command of the specified command type executed during the one-hour period analyzed by the report.
  - The **Min Duration(sec)** column displays the shortest amount of time (in seconds) used by the server to respond to the specified command type.



- The **Max Duration(sec)** column displays the longest amount of time (in seconds) used by the server to respond to the specified command type.
  - The **Avg Duration(sec)** column displays the average length of time (in seconds) used by the server when responding to the specified command type.
- The **DML Statistics Timeline** section of the Log Analysis Expert report displays information about DML statement usage:
  - The line graph displays an analysis of statement usage during the selected time period. Hover over a specific point to view detailed information about that point on the graph.
  - The pie chart displays the percent of statement usage of each respective DML statement type during the selected time period.
- The **DDL Statistics Timeline** section of the Log Analysis Expert report displays information about DDL statement usage:
  - The line graph displays an analysis of statement usage during the selected time period. Hover over a specific point to view detailed information about that point on the graph.
  - The pie chart displays the percent of statement usage of each respective DDL statement type during the selected time period.
- The **Commit and Rollback Statistics Timeline** section of the Log Analysis Expert report displays information about the COMMIT, ROLLBACK, and SAVEPOINT statements logged during the specified time period:
  - The line graph displays an analysis of the commit and rollback activity during the specified time period. Hover over a specific point to view detailed information about that point on the graph.
  - The pie chart displays the comparative percent of COMMIT, SAVEPOINT, or ROLLBACK statements executed during the specified time period.
- The **Checkpoint Statistics Timeline** section of the Log Analysis Expert report displays information about the checkpoint operations logged during the specified time period:
  - The line graph displays an analysis of the checkpoint operation activity during the specified time period. Hover over a specific point to view detailed information about that point on the graph.
  - The pie chart displays the comparative percent of different types of checkpoint activity logged during the specified time period.
- The **Log Event Statistics** table lists log entries with a severity level of WARNING, ERROR, FATAL, PANIC, HINT or CONTEXT. The level of logging detail for error messages is controlled by the **log\_min\_error\_statement** parameter. You can set the parameter by either modifying the **postgresql.conf** file with your editor of choice, or by specifying a value in the **Log Min Error Statement** field of the **Log Manager** wizard.
  - The **Error Severity** column lists the severity level of the log entry.
  - The **Message** column lists the log message.
  - The **Total Count** column lists the number of times that the log entry has occurred.
- The **Log Statistics** table lists log entries that indicate an operational severity level of LOG, DETAIL, DEBUG, NOTICE, INFO or STATEMENT. The level of logging detail for informational messages is controlled by the **log\_min\_messages** parameter. You can set the parameter by either modifying the **postgresql.conf** file with your editor of choice, or by specifying a value in the **Log Min Messages** field of the **Log Manager** wizard.
  - The **Error Severity** column lists the severity level of the log entry.
  - The **Total Count** column lists the number of times that the log entry has occurred.
- The **Temp Generated Queries** table displays a list of queries that have created temporary files.
  - The **Log Time** column displays the time that the log entry was generated.
  - The **TempFile Size(Bytes)** column displays the size of the temporary file in bytes.
  - The **Query** column displays the text of the query that created the temporary file.
- The **Temp File Statistics Timeline** graph displays the size of temporary files over the specified time period. Hover over a specific point to view detailed information about that point on the graph.
- The **Lock Statistics Timeline** section of the Log Analysis Expert report displays information about the locks held during the specified time period:
  - The graph displays the number of locks held at any given point during the time period. Hover over a specific point to view detailed information about that point on the graph.
  - The pie chart displays the relative percentage of each type of lock used during the selected time period.
- The **Waiting Statistics Timeline** section of the Log Analysis Expert report displays information about DML statements that are waiting for a lock during the specified time period:
  - The graph displays the number of DML statements that are waiting at any given point during the time period; each colored line represents a statement type. Hover over a specific point to view detailed information about that point on the graph.
  - The pie chart displays the relative percentage of each type of DML statement that waited for a lock during the selected time period.

- The **Idle Statistics Timeline** section of the Log Analysis Expert report displays information about the amount of time that a connection to the server is idle. An **IDLE** server is waiting for a connection from a client. A connection that is **IDLE in transaction** has started a transaction, but has not yet committed or rolled back the transaction and is waiting for a command from the client. A session that is **IDLE in transaction (aborted)\*** has started a transaction, but has not yet committed or rolled back the transaction and is waiting for a command from the client; an error has occurred within the transaction and the transaction can only be rolled-back.
  - The graph displays the times at which the server is **IDLE**, **IDLE in transaction**, and **IDLE in transaction (aborted)**. Hover over a specific point to view detailed information about that point on the graph.
  - The pie chart displays the relative percentage of each type of lock used during the selected time period.
- The **Autovacuum Statistics** table displays statistics about autovacuum activity on monitored servers.
  - The **Log Time** column displays the time that the autovacuum activity was written to the log.
  - The **Relation** column displays the name of the table on which the autovacuum was performed.
  - The **Index Details** column displays the number of index scans that were performed.
  - The **Page Details** column displays the number of pages that were removed, and the number of pages that remain.
  - The **Tuple Details** column displays the number of tuples that were removed, and the number of tuples that remain.
  - The **Buffer Usage** column displays the number of buffers hit, missed, or dirty.
  - The **Read Rate** column displays the average read rate in MB's per second.
  - The **System Usage** column displays the percent of CPU time used performing autovacuum activities.
- The **Autoanalyze Statistics** table displays logged autoanalyze activity.
  - The **Log Time** column displays the time that the autoanalyze activity was written to the log.
  - The **Relation** column displays the name of the table on which the autoanalyze was performed.
  - The **System Usage** column displays the percent of CPU time used performing autoanalyze activities.
- The **Slow Query Statistics** table displays the slowest queries executed on monitored servers. The table will include the number of entries specified in the **Rows Limit** field of the Log Analysis Expert.
  - The **Log Time** column displays the time that the query activity was written to the log.
  - The **Tag** column displays the command type.
  - The **Query** column displays the text of the performed query.
  - The **Parameters** column displays the parameters (if the query is a parameterized query).
  - The **Duration** column displays the length of time that it took the server to execute the query.
  - The **Host** column displays name of the host on which the query executed.
  - The **Database** column displays the name of the database on which the query executed.
- The **Frequently Executed Query Statistics** table displays the most frequently executed query statements. The table will include the number of entries specified in the **Rows Limit** field of the Log Analysis Expert.
  - The **Query** column displays the text of the performed query.
  - The **Parameters** column displays the parameters (if the query is a parameterized query).
  - The **No. of Times Executed** column displays the number of times that the query executed.
  - The **Total Duration** column displays the length of time that it took the server to execute the query.
- The **Most Time Executed Query Statistics** table displays the queries that took the most execution time on the server. The table will include the number of entries specified in the **Rows Limit** field of the Log Analysis Expert.
  - The **Query** column displays the text of the performed query.
  - The **Parameters** column displays the parameters (if the query is a parameterized query).
  - The **No. of Times Executed** column displays the number of times that the query executed.
  - The **Total Duration** column displays the length of time that it took the server to execute the query.
- The **Connections Overview Timeline** section of the Log Analysis Expert report displays information about successful and unsuccessful connection attempts during the specified time period:
  - The **Timestamp** graph displays the number of server connections attempted and connections authenticated at any given point during the specified time period. Hover over a specific point to view detailed information about that point on the graph.
  - The **Summary** pie chart displays the relative percentage of connections attempted and connections authenticated during the specified time period.

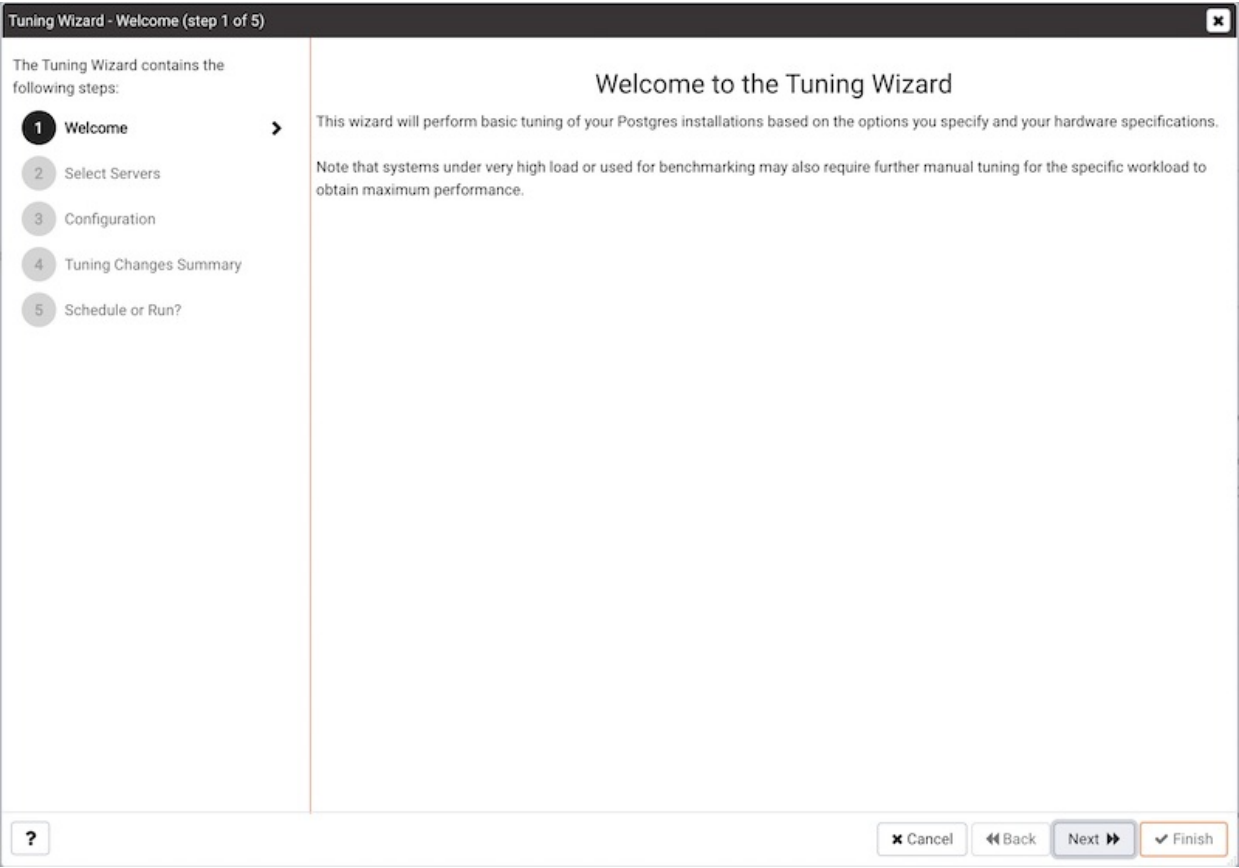
## 34.4.6 Tuning Wizard

The Tuning Wizard reviews your PostgreSQL or Advanced Server installation, and recommends a set of configuration options that will help tune the installation to best suit its anticipated workload. Please note that benchmarking systems or systems with a high work load may require additional manual tuning to reach optimum performance.

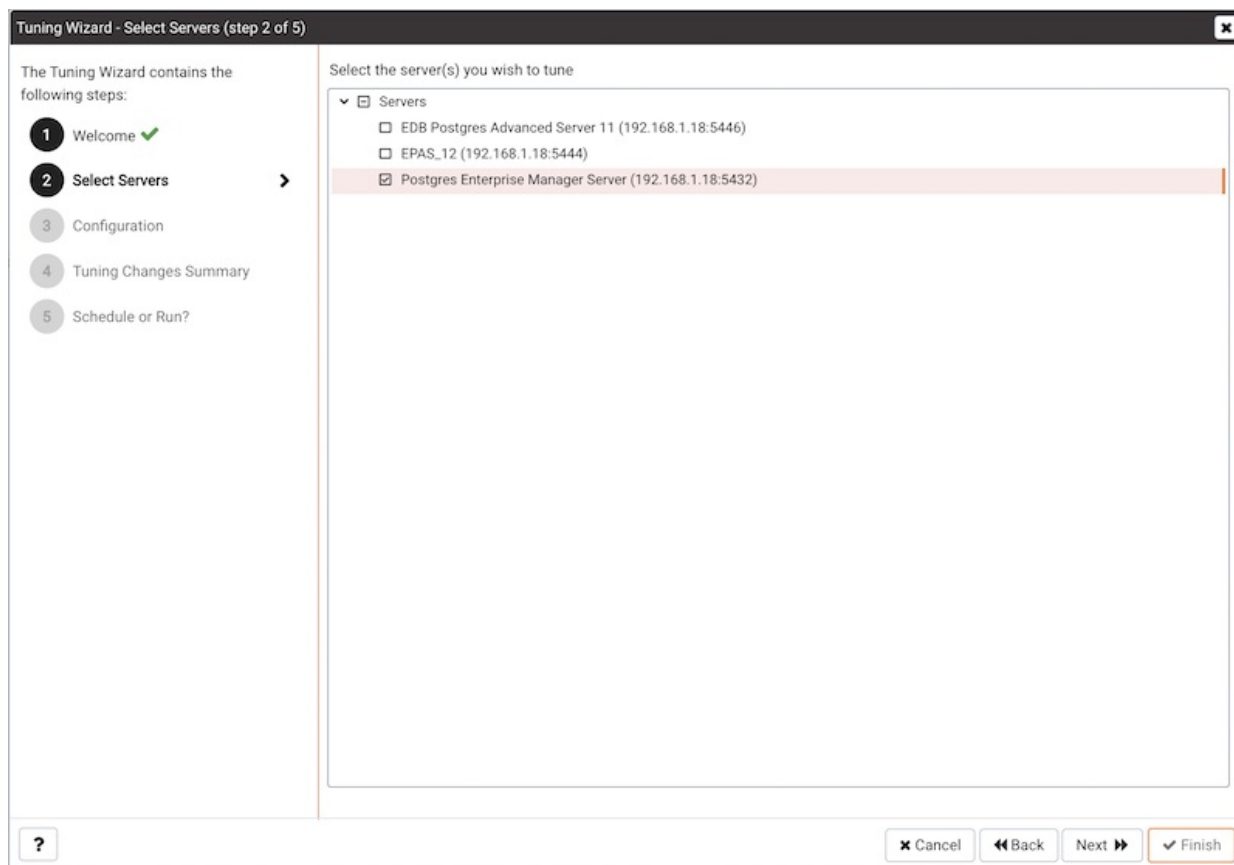
Before using the Tuning Wizard, you must specify the name of the service in the **Service ID** field on the **Advanced** tab of the server's **Properties** dialog. PEM will use the service name when restarting the service after tuning.

The Tuning Wizard can only make recommendations for those servers that reside on the same server as their bound PEM agent. If you have specified a value of **Yes** in the **Remote monitoring** field when defining your server, the server will not be displayed in the Tuning Wizard tree control.

To open the Tuning Wizard, select **Tuning Wizard...** from the **Management** menu of the PEM client. The Tuning Wizard opens, welcoming you:



Click **Next** to continue to the server selection dialog:



Expand the **Servers** node of the tree control to view a list of the servers that are currently monitored by PEM that are available for tuning.

Check a box to the left of a server name to select the server for tuning. Please note: the Tuning Wizard displays a red warning symbol to the left of a server name in the tree control if the service name for that server is not provided on the server's Properties dialog.

Click **Next** to continue to the **Configuration** dialog:

**Tuning Wizard - Configuration (step 3 of 5)**

The Tuning Wizard contains the following steps:

- 1 Welcome ✓
- 2 Select Servers ✓
- 3 Configuration >
- 4 Tuning Changes Summary
- 5 Schedule or Run?

**Machine utilization**

☒ Dedicated
 ☐ Mixed use
 ☐ Developer workstation

This machine is dedicated to run Postgres and will use available memory to optimize performance.

**Workload selection**

☒ OLTP
 ☐ Mixed
 ☐ Data warehouse

The running application is transaction-intensive application.

Buttons: ? Cancel Back Next Finish

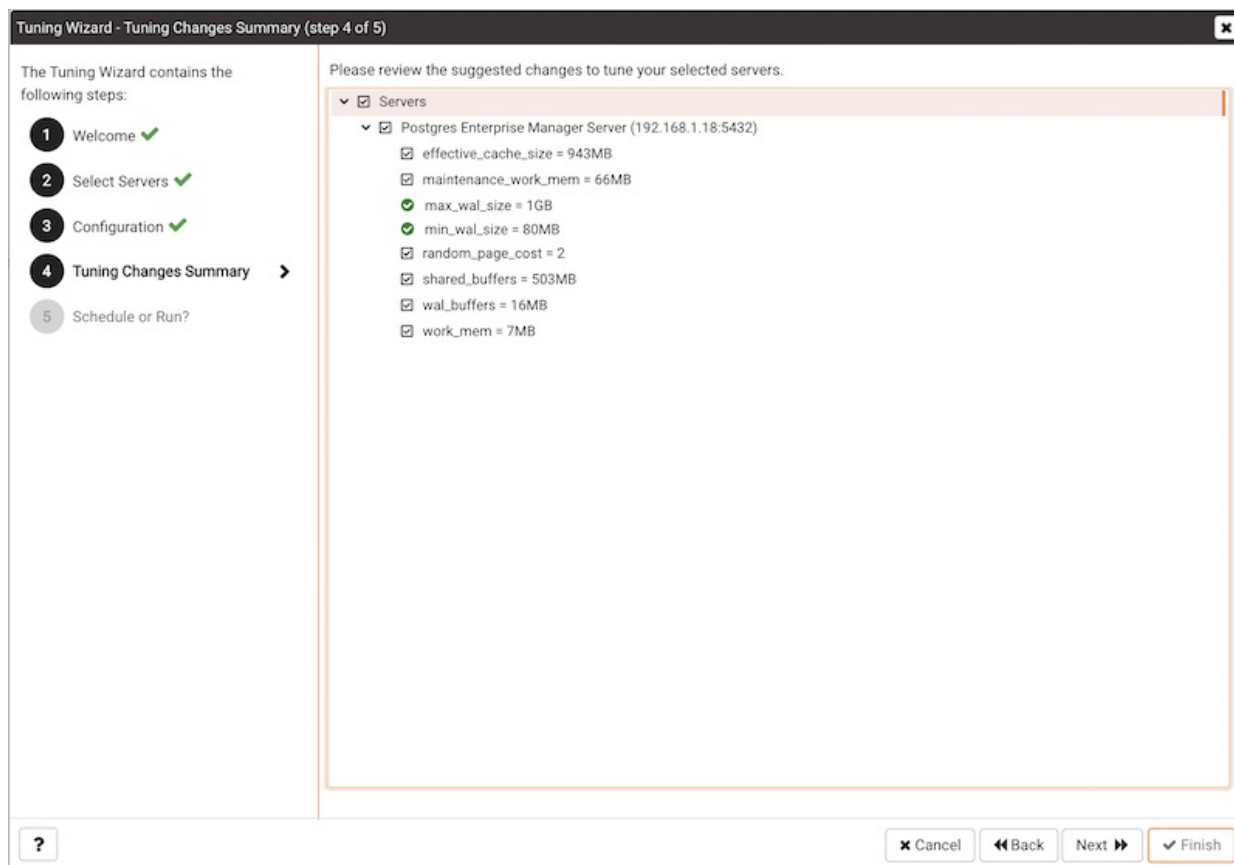
Select an option in the **Machine utilization** field to specify the type of work performed by the selected servers. The type of work performed by the server determines how the tuning wizard will allocate system resources:

- Select **Dedicated** to dedicate the majority of the system resources to the database server.
- Select **Mixed use** to dedicate a moderate amount of system resources to the database server.
- Select **Developer workstation** to dedicate a relatively small amount of system resources to the database server.

Select an option in the **Workload Selection** field to specify the type of workload typically performed on the selected server:

- Select **OLTP** if the selected server is used primarily to process online transaction workloads.
- Select **Mixed** if the selected server provides a mix of transaction processing and data reporting.
- Select **Data warehouse** if the server is used for heavy data reporting.

Click **Next** to continue to the **Tuning Changes Summary** dialog:



The tree control on the **Tuning Changes Summary** dialog displays the parameter setting modifications recommended for each server analyzed by the Tuning Wizard. Use the checkboxes next to a server or parameter name to select the recommendations that tuning wizard will either include in a preview report or apply:

- A checked box to the left of a parameter name specifies that the Tuning Wizard will include the parameter setting.
- A checked box to the left of a server name specifies that the Tuning Wizard will include all parameter setting recommendations for the specified server.

Specify which Tuning Wizard recommendations you wish to include in a report or apply, and click **Next** to continue.

Use the **Schedule or Run?** dialog to either:

- Specify a time that PEM will apply the changes.
- Generate a report that details the recommended changes.

The selected actions will apply to all of the changes noted on the **Tuning Changes Summary**. If you opt to generate a report, PEM will create a report that contains a list of the current values and recommended modifications to the configuration parameters selected on the **Tuning Changes Summary** dialog. Note that to implement changes, you will need to invoke the Tuning Wizard a second time, specifying the parameters you wish to modify on the **Tuning Changes Summary** dialog.

Select **Schedule changes** to view and specify your scheduling options.

Tuning Wizard - Schedule or Run? (step 5 of 5)


The Tuning Wizard contains the following steps:

- 1 Welcome ✓
- 2 Select Servers ✓
- 3 Configuration ✓
- 4 Tuning Changes Summary ✓
- 5 Schedule or Run? >

**Schedule/Generate Tuning Changes**

☒ Schedule changes ☐ Generate report

Configure now? ☒ Yes ☐ No

Time?  

NOTE: The selected servers will be restarted to complete tuning operation.

?

You can:

- Set the **Configuration now?** slider to **Yes** to apply the tuning wizard's recommendations and restart/reload the server now.
- Set the **Configuration now?** slider to **No** to enable the **Time?** field and use the calendar selector to specify a time for PEM to apply the tuning wizard's recommendations and restart/reload the server. Note that if you schedule a time for the changes to be applied, you will not be provided with a preview of the change recommendations.

Select **Generate report** to view your report options.

**Tuning Wizard - Schedule or Run? (step 5 of 5)**

The Tuning Wizard contains the following steps:

- 1 Welcome ✓
- 2 Select Servers ✓
- 3 Configuration ✓
- 4 Tuning Changes Summary ✓
- 5 Schedule or Run? ➤

**Schedule/Generate Tuning Changes**

Schedule changes

View report now?

Save the report to file

Note: Filename can only be latin-1 compatible characters.

Buttons:

You can:

- Set the **View report now?** slider to **Yes** to display the Tuning Wizard report onscreen.
- Set the **View report now?** slider to **No** to enable the **Save the report to file** field and use the calendar selector to specify a file name and location to which PEM will write the Tuning Wizard report.

Click the **Finish** button to either apply the Tuning Wizard's modifications or generate a report and exit the Tuning Wizard.

| Tuning Wizard Report  |                |   |
|---|----------------|---|
| Generated On: 2020-04-27 16:19:52   |                | Go to: Postgres Enterprise Manager Server |
| Summary   |                |   |
| Number of servers selected: 1 Machine utilization: Dedicated Workload profile: OLTP |                |   |
| Server: Postgres Enterprise Manager Server (192.168.1.18:5432)                      |                |   |
| GUC Parameter   | Original Value | Recommended Value                         |
| effective_cache_size  | 4096MB         | 943MB                                     |
| maintenance_work_mem  | 64MB           | 66MB                                      |
| random_page_cost  | 4              | 2   |
| shared_buffers  | 128MB          | 503MB                                     |
| wal_buffers   | 4MB            | 16MB                                      |
| work_mem  | 4MB            | 7MB                                       |

You can confirm that Tuning Wizard has implemented the recommended changes by reviewing the postgresql.conf file for the modified server. The Tuning Wizard adds a comment above each modified parameter in the postgresql.conf file when the change is applied:



```

root@localhost:/opt/PostgresPlus/9.5AS/data
File Edit View Search Terminal Help
# The value for shared_buffers was changed by the Postgres Enterprise Manager Tuning Wizard at 2017-03-06 09:15:32
#shared_buffers = 196MB
# The value for shared_buffers was changed by the Postgres Enterprise Manager Tuning Wizard at 2017-03-06 09:18:23
shared_buffers = 196MB

# (change requires restart)
# on, off, or try
# (change requires restart)
# min 800kB
# zero disables the feature
# (change requires restart)
#huge_pages = try

#temp_buffers = 8MB
#max_prepared_transactions = 0

# Note: Increasing max_prepared_transactions costs ~600 bytes of shared memory
# per transaction slot, plus lock space (see max_locks_per_transaction).
# It is not advisable to set max_prepared_transactions nonzero unless you
# actively intend to use prepared transactions.
#work_mem = 4MB # min 64kB
# The value for work_mem was changed by the Postgres Enterprise Manager Tuning Wizard at 2017-03-06 09:15:32
#work_mem = 3MB
# The value for work_mem was changed by the Postgres Enterprise Manager Tuning Wizard at 2017-03-06 09:18:23
work_mem = 3MB
#maintenance_work_mem = 64MB # min 1MB
# The value for maintenance_work_mem was changed by the Postgres Enterprise Manager Tuning Wizard at 2017-03-06 09:15:32
#maintenance_work_mem = 25MB
--More-- (21%)

```

You can also confirm a parameter value by querying the server. For example, to confirm the value of the `shared_buffers` parameter, open a SQL command line using either the [Query tool](#) (accessed through the [Tools](#) menu) or the `psql` client, and issue the command:

```
SHOW shared_buffers;
```

The value returned by the server will confirm that the parameter has been modified.

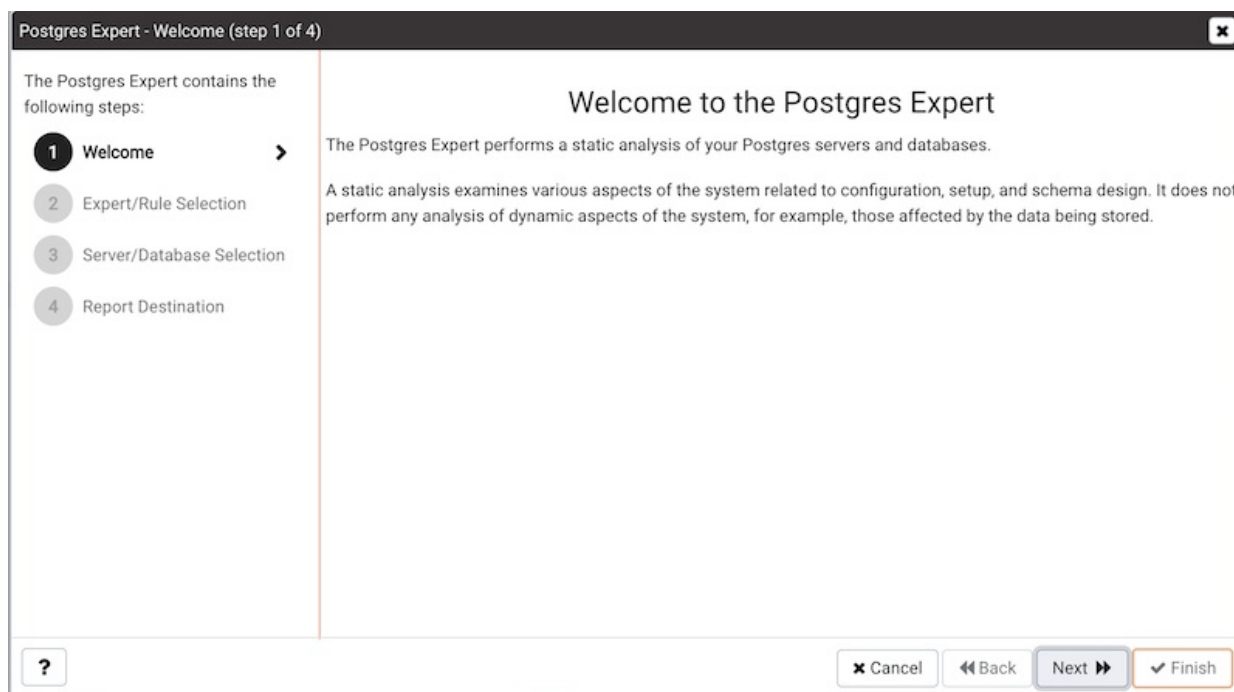
## 34.4.7 Postgres Expert

Postgres Expert analyzes the configuration of servers that are registered with the Enterprise Manager, and provides advice about:

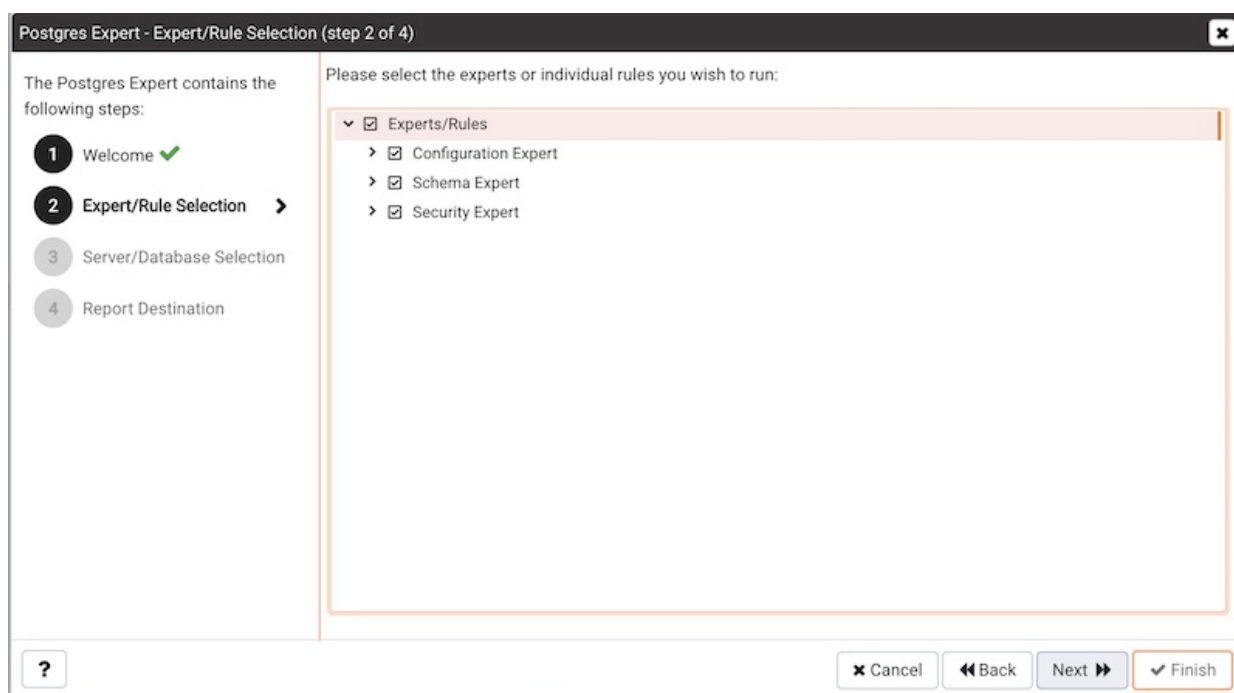
- [Server Performance](#)
- [Server Security](#)
- [Server Configuration](#)

Postgres Expert is an advisory utility; after analyzing the selected servers, Postgres Expert produces a report containing analysis of potential performance and security issues, along with suggestions for addressing each such issue.

To use the Postgres Expert wizard select the [Postgres Expert](#) option from the [Management](#) menu in the PEM client. When the wizard's [Welcome](#) window opens; click [Next](#) to continue:



The wizard displays a tree control that allows you to choose the **Experts** and **Rules** with which Postgres Expert will evaluate the specified server or database.



The tree control categorizes the Rules under three Expert headings:

- Select from the **Configuration Expert** rules to analyze the parameter settings of the server or operating system to find any adjustments that might improve system performance.
- Select from the **Schema Expert** rules to analyze schema objects (locating missing primary keys, foreign keys without indexes, etc).
- Select from the **Security Expert** rules to review the system to find security vulnerabilities.

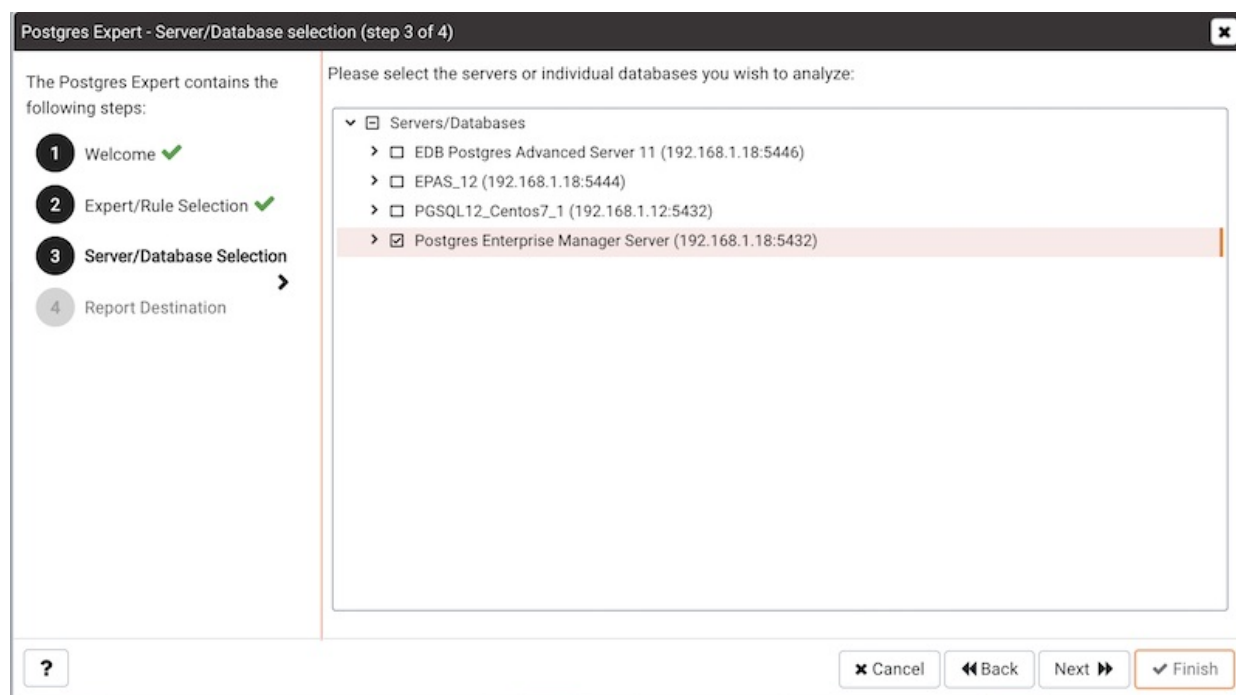
Use the checkbox to the left of an expert or rule to indicate that the Postgres Expert should analyze the configuration of the selected servers for any best practice deviations related to the selected item.

- Use the checkbox next to **Experts/Rules** to select or deselect all of the items listed in the tree control.
- Use the checkbox next to the name of an expert to select or deselect all of the configuration items listed under that node of the tree

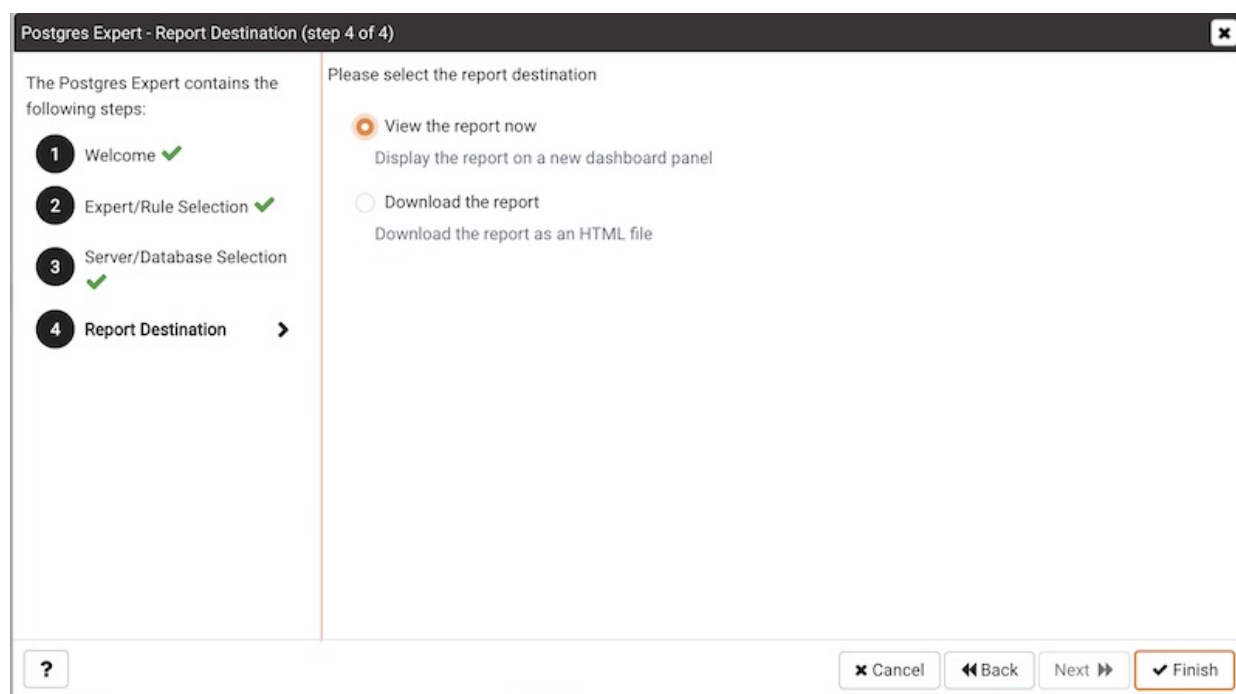
control.

- Use the checkbox next to a rule to select or deselect the rule for inclusion in the Postgres Expert report.

After making your selections, click **Next** to continue to the **Server/Databases** tree control.



If you select multiple servers or databases, the resulting report will contain a separate analysis of each target. Select or de-select the servers and databases that you would like Postgres Expert to analyze, and select **Next** to continue.




You can select the default option and click **Finish** to immediately view an onscreen report from Postgres Expert, or check the box next to **Download the report** to save a copy of the report to an HTML file for later use. If you choose to download the report, the report will be saved in your default downloads directory.

## Reviewing the Postgres Expert Report

If you've elected to review the report immediately, the PEM client will display the report on the **Postgres Expert** tab.

Postgres Expert Report

 Generated On: 2020-04-27 16:57:47

Go to: Postgres Enterprise Manager Server

Summary

Servers Tested: 1 Rules Checked: 31 High Alerts: 1 Medium Alerts: 3 Low Alerts: 2

Server: Postgres Enterprise Manager Server (192.168.1.18:5432)

Advisor: Configuration Expert

| Rule                               | Database | Severity |
|------------------------------------|----------|----------|
| Check checkpoint_completion_target | -        | Medium   |
| Check effective_cache_size         | -        | Medium   |
| Check effective_io_concurrency     | -        | Low      |
| Check reducing_random_page_cost    | -        | Low      |

Advisor: Schema Expert

| Rule   | Database | Severity |
|--|----------|----------|
| Check data and transaction log on same drive | -        | High     |
| Check for missing foreign key indexes        | db01     | Medium   |


A report summary in the upper-left corner of the Postgres Expert Report lists statistics about the analysis, including the number of servers analyzed, the number of rules tested, and the number of alerts raised in each severity category.

If your report contains recommendations for more than one server, you can use the **Jump to** selector in the upper-right corner of the report as a navigation tool; select a server from the list to move to the portion of the report containing information for the selected server.

For each server analyzed, the Postgres Expert returns recommendations from the **Configuration Expert**, the **Schema Expert**, and the **Security Expert**. Each expert returns a list of rules that raised an alert, the database that the rule pertains to, and the severity level of the alert. Click on a rule name to view detailed information about the selected rule:

|                   |   |
|-------------------|---|
| Section Heading   | Contains  |
| Trigger           | A description of the rule that raised the alert.                              |
| Recommended Value | The value to which Postgres Expert recommends setting the selected parameter. |
| Description       | Information and advice about the parameter that caused the alert.             |
| Current Values    | The current value(s) of the parameter(s).                                     |

Postgres Expert Report

 Generated On: 2020-04-27 16:57:47

Go to: Postgres Enterprise Manager Server

Summary

Servers Tested: 1 Rules Checked: 31 High Alerts: 1 Medium Alerts: 3 Low Alerts: 2

Server: Postgres Enterprise Manager Server (192.168.1.18:5432)

Advisor: Configuration Expert

| Rule                               | Database | Severity |
|------------------------------------|----------|----------|
| Check checkpoint_completion_target | -        | Medium   |

Recommended Value: Consider adjusting checkpoint\_completion\_target.

Current Values:

| Settings                     | Value |
|------------------------------|-------|
| checkpoint_completion_target | 0.5   |

Trigger: checkpoint\_completion\_target != 0.9

Description: In order to ensure reliable and efficient crash recovery, PostgreSQL periodically writes all dirty buffers to disk. This process is called a checkpoint. Beginning in PostgreSQL 8.3, checkpoints take place over an extended period of time in order to avoid swamping the I/O system. checkpoint\_completion\_target controls the rate at which the checkpoint is performed, as a function of the time remaining before the next checkpoint is due to start. A value of 0 indicates that the checkpoint should be performed as quickly as possible, whereas a value of 1 indicates that the checkpoint should complete just as the next checkpoint is scheduled to start. It is usually beneficial to spread the checkpoint out as much as possible; however, if checkpoint\_completion\_target is set to a value greater than 0.9, unexpected delays near the end of the checkpoint process can cause the checkpoint to fail to complete before the next one needs to start. Because of this, the recommended setting is 0.9.

Check effective\_cache\_size

-

Medium

Check effective\_io\_concurrency

-

Low

Check reducing\_random\_page\_cost

-

Low

Advisor: Schema Expert

| Rule   | Database | Severity |
|--|----------|----------|
| Check data and transaction log on same drive | -        | High     |
| Check for missing foreign key indexes        | db01     | Medium   |

For more information about each rule checked by the Postgres Expert, see:

### 34.4.7.1 Schema Expert Recommendations

|                |   |
|----------------|---|
| Rule           | Check for missing primary keys                                |
| Recommendation | Ensure tables have a primary key                              |
| Trigger        | Postgres Expert detected a table with no defined primary key. |
| Severity       | Low   |

**Description:** Primary keys are used to define the set of columns that make up the unique key to each row in the table. Whilst they are similar to unique indexes, primary keys cannot contain NULL values, thus are always able to identify a single row. Tools such as Postgres Enterprise Manager and other pieces of software such as ORM will automatically detect primary keys on tables and use their definition to identify individual rows.

|                |  |
|----------------|--|
| Rule           | Check for missing foreign key indexes  |
| Recommendation | Ensure columns of child tables in foreign key relationships are indexed.       |
| Trigger        | Postgres Expert detected a child table with no index on referencing column(s). |
| Severity       | Medium   |

**Description:** Foreign keys are used to define and enforce relationships between child and parent tables. The foreign key specifies that values in one or more columns of the child table must exist (in the same combination, if more than one column) in the referenced column(s) of the parent table. A unique index is required to be present on the referenced columns in the parent table, however an index is not required, but is generally advisable, on the referencing columns of the child table to allow cascading updates to the parent to be executed efficiently.

|                |   |
|----------------|---|
| Rule           | Check Database Encoding                   |
| Recommendation | Avoid encoding as SQL_ASCII for databases |
| Trigger        | encoding = SQL_ASCII                      |
| Severity       | Medium                                    |

**Description:** The database is created to store data using the SQL\_ASCII encoding. This encoding is defined for 7 bit characters only; the meaning of characters with the 8th bit set (non-ASCII characters 127-255) is not defined. Consequently, it is not possible for the server to convert the data to other encodings. If you're storing non-ASCII data in the database, you're strongly encouraged to use a proper database encoding representing your locale character set to take benefit from the automatic conversion to different client encodings when needed. If you store non-ASCII data in an SQL\_ASCII database, strange characters may be written to or read from the database, caused by code conversion problems. This may cause problems when accessing the database using different client programs and drivers. For most installations, Unicode (UTF8) encoding will provide the most versatility.

|                |   |
|----------------|---|
| Rule           | Check for too many indexes  |
| Recommendation | Don't overload a table with too many indexes.                       |
| Trigger        | Postgres Expert has detected that a table has more than 10 indexes. |
| Severity       | Low, Medium or High (based on number of indexes)                    |

**Description:** Whilst indexes can speed up SELECT queries by allowing Postgres to quickly locate records, it is important to choose which indexes are required carefully to ensure they are used. Maintaining an index has a cost, and the more indexes there are to update, the slower INSERT, UPDATE or DELETE queries can become. There are no hard and fast rules to tell you how many indexes are required on a particular table -the DBA must balance the need for indexes for different types of SELECT queries and constraints against the cost of maintaining them.

|                    |  |
|--------------------|--|
| Configuration Item | Check data and transaction log on same drive   |
| Recommendation     | Avoid using the same storage device for the data directory and transaction logs.                 |
| Trigger            | Postgres Expert has detected that a data directory and transaction log directory share a device. |
| Severity           | High   |

**Description:** Postgres' performance can be adversely affected on medium to heavily loaded systems if both the data and the transaction logs (WAL) are stored on the same device. It is considered good practice to store them on separate physical devices if performance is an issue. On busy servers, significant performance gains may be seen when separating the data directory and transaction log directory onto different physical storage devices.

|                |  |
|----------------|--|
| Rule           | Check tablespace and transaction log on same drive   |
| Recommendation | Avoid using the same storage device for the transaction logs and a tablespace.                                     |
| Trigger        | Postgres Expert has detected that transaction log directory and a tablespace other than pg_default share a device. |
| Severity       | Medium   |

**Description:** Before updating database files to reflect data modifications, the server writes the change to the transaction log. The database files may be separated onto different devices using tablespaces (defined storage areas used by the database server). On busy servers, significant performance gains may be seen when separating tablespace directories and the transaction log directory onto different physical storage devices.

|                |  |
|----------------|--|
| Rule           | Check multiple tablespace on same drive                                |
| Recommendation | Avoid using the same storage device for multiple tablespaces.          |
| Trigger        | Postgres Expert has detected that multiple tablespaces share a device. |
| Severity       | Low  |

**Description:** Multiple tablespaces may be defined in the database to allow tables and indexes to be distributed into different storage areas, usually for performance reasons for example, tables with high performance requirements may be stored on expensive , high speed disks, while archive data may be stored on much larger, but slower devices. There is usually little to be gained from having more than one tablespace on a single device (because the cost and access characteristics will be identical), except in very unusual situations where it may be desirable to configure them with different planner cost parameters.

### 34.4.7.2 Security Expert Recommendations

|                |  |
|----------------|--|
| Rule           | Check SSL for improved performance                                 |
| Recommendation | Consider disabling SSL for improved performance.                   |
| Trigger        | ssl = on and listen_addresses in ('localhost', '127.0.0.1', '::1') |
| Severity       | Low  |

**Description:** SSL authentication is invaluable for protecting against connection-spoofing and eavesdropping attacks, but it is not always necessary for adequate security. When PostgreSQL accepts only local connections, or when it accepts only connections from a trusted network where malicious network traffic is not a concern, SSL encryption may not be necessary. Consider changing this setting if the current value is not appropriate for your environment.

Note: Even when SSL encryption is enabled, PostgreSQL servers should be further protected using an appropriate firewall configuration.

|                |   |
|----------------|---|
| Rule           | Check SSL for improved connection security                              |
| Recommendation | Consider using SSL for improved connection security.                    |
| Trigger        | ssl = off and listen_addresses not in ('localhost', '127.0.0.1', '::1') |
| Severity       | Medium  |

**Description:** The configuration variable listen\_addresses indicates that your system may accept non-local connection requests, but SSL is not enabled. If PostgreSQL is exposed only to a secure, trusted internal network, this configuration is appropriate for maximum performance. Otherwise, you should consider enabling SSL. SSL offers two main advantages. First, it provides a more secure mechanism for authorizing connections to the database, helping to prevent unauthorized access. Second, SSL prevents eavesdropping attacks, where data sent from the database to clients, or from clients to the database, is viewed by an attacker while in transit. Consider changing this setting if the current value is not appropriate for your environment.

Note: Even when SSL encryption is enabled, PostgreSQL servers should be further protected using an appropriate firewall configuration.

|                |   |
|----------------|---|
| Rule           | Check TRUST authentication is disabled  |
| Recommendation | Avoid trust and ident authentication on unsecured networks.                   |
| Trigger        | trust or ident authentication allowed to any host other than 127.0.0.1 or ::1 |
| Severity       | High  |

**Description:** An attacker with access to your network can easily use the trust and ident authentication methods to subvert your network. If PostgreSQL is not running on a secure network, with firewalls in place to prevent malicious traffic, the use of these authentication methods should be avoided.

|                |  |
|----------------|--|
| Rule           | Check Password authentication on unsecured networks                                |
| Recommendation | Avoid password authentication on unsecured networks.                               |
| Trigger        | (connection_type = 'host' or connection_type = 'hostnssl') and method = 'password' |
| Severity       | High   |

**Description:** Passwords should not be transmitted in plaintext over unsecured networks. The use of md5 authentication provides slightly better security,

but can still allow accounts to be compromised by a determined attacker. SSL encryption is a superior alternative. To require the use of SSL, set the connection type to hostssl in the pg\_hba.conf file.

|                |  |
|----------------|--|
| Rule           | Check SSL for increased security                                 |
| Recommendation | Consider requiring SSL.  |
| Trigger        | ssl = on in postgresql.conf, but no hostssl lines in pg_hba.conf |
| Severity       | Medium   |

**Description:** SSL encrypts passwords and all data transmitted over the connection, providing increased security. To require the use of SSL, set the connection type to hostssl in the pg\_hba.conf file.

### 34.4.7.3 Configuration Expert Recommendations

|                   |   |
|-------------------|---|
| Rule              | Check shared_buffers  |
| Recommendation    | Consider adjusting shared_buffers   |
| Trigger           | shared_buffers < (OS == Windows ? 64MB : MIN(0.20 * (system_memory - 256MB), 6GB)) or shared_buffers > (OS == Windows ? 512MB : MAX(0.35 * system_memory, 8GB)) |
| Recommended Value | system_memory < 1GB ? MAX((system_memory - 256MB) / (OS == Windows ? 6 : 3), 64MB), OS == Windows ? MAX(system_memory / 8, 256MB) : MAX(system_memory / 4, 8GB) |
| Severity          | Medium  |

**Description:** The configuration variable shared\_buffers controls the amount of memory reserved by PostgreSQL for its internal buffer cache. Setting this value too low may result in "thrashing" the buffer cache, resulting in excessive disk activity and degraded performance. However, setting it too high may also cause performance problems. PostgreSQL relies on operating system caching to a significant degree, and setting this value too high may result in excessive "double buffering" that can degrade performance. It also increases the internal costs of managing the buffer pool. On UNIX-like systems, a good starting value is approximately 25% of system memory, but not more than 8GB. On Windows systems, values between 64MB and 512MB typically perform best. The optimal value is workload-dependent, so it may be worthwhile to try several different values and benchmark your system to determine which one delivers best performance.

Note: PostgreSQL will fail to start if the necessary amount of shared\_memory cannot be located. This is usually due to an operating system limitation which can be raised by changing a system configuration setting, often called shmall. See the documentation for more details. You must set this limit to a value somewhat higher than the amount of memory required for shared\_buffers, because PostgreSQL's shared memory allocation also includes amounts required for other purposes.

|                   |  |
|-------------------|--|
| Rule              | Check work_mem   |
| Recommendation    | Consider adjusting work_mem  |
| Trigger           | given spare_mem = system_memory - (OS == Windows ? 256MB : MAX(0.25 * system_memory, 8GB)) then work_mem < MAX(1MB, spare_mem / 512) or work_mem > (spare_mem / 128) |
| Recommended Value | given spare_mem defined as on the previous line then MAX (1MB, spare_mem / 256)  |
| Severity          | Medium   |

**Description:** The configuration variable work\_mem controls the amount of memory PostgreSQL will use for each individual hash or sort operation. When a sort would use more than this amount of memory, the planner will arrange to perform an external sort using disk files. While this algorithm is memory efficient, it is much slower than an in-memory quick sort. Similarly, when a hash join would use more than this amount of memory, the planner will arrange to perform it in multiple batches, which saves memory but is likewise much slower. In either case, the planner may in the alternative choose some other plan that does not require the sort or hash operation, but this too is often less efficient. Therefore, for good performance it is important to set this parameter high enough to allow the planner to choose good plans. However, each concurrently executing query can potentially involve several sorts or hashes, and the number of queries on the system can vary greatly. Therefore, a value for this setting that works well when the system is lightly loaded may result in swapping when the system becomes more heavily loaded. Swapping has very negative effects on database performance and should be avoided, so it is usually wise to set this value somewhat conservatively.



Note: `work_mem` can be adjusted for particular databases, users, or user-and -database combinations by using the commands `ALTER ROLE` and `ALTER DATABASE`. It can also be changed for a single session using the `SET` command. This can be helpful when particular queries can be shown to run much faster with a value of `work_mem` that is too high to be applied to the system as a whole.

|                |                                       |
|----------------|---------------------------------------|
| Rule           | Check <code>max_connections</code>    |
| Recommendation | Consider using a connection pooler    |
| Trigger        | <code>max_connections &gt; 100</code> |
| Severity       | Medium                                |

**Description:** The configuration variable `max_connection` is set to a value greater than 100. PostgreSQL performs best when the number of simultaneous connections is low. Peak throughput is typically achieved when the connection count is limited to is limited to approximately twice the number of system CPU cores plus the number of spindles available for disk I/O (in the case of an SSD or other non-rotating media, some experimentation may be needed to determine the "effective spindle count"). Installing a connection pooler, such as `pgpool-II` or `pgbouncer`, can allow many clients to be multiplexed onto a smaller number of server connections, sometimes resulting in dramatic performance gains.

|                   |   |
|-------------------|---|
| Rule              | Check <code>maintenance_work_mem</code>   |
| Recommendation    | Consider adjusting <code>maintenance_work_mem</code>  |
| Trigger           | <code>spare_mem = system_memory - (OS == Windows ? 256MB : MAX(0.25 * system_memory, 8GB))</code> then <code>maintenance_work_mem &lt; MAX(16MB, spare_mem / 32)</code> or <code>maintenance_work_mem &gt; MIN(spare_mem / 8, 256MB)</code> |
| Recommended Value | <code>spare_mem</code> as defined on the previous line then <code>MIN(spare_mem/16, 256MB)</code>   |
| Severity          | Low   |

**Description:** The configuration variable `maintenance_work_mem` controls the amount of memory PostgreSQL will use for maintenance operations such as `CREATE INDEX` and `VACUUM`. Increasing this setting from the default of 16MB to 256MB can make these operations run much faster. Higher settings typically do not produce a significant further improvement. On PostgreSQL 8.3 and higher, multiple autovacuum processes may be running at one time (up to `autovacuum_max_workers`, which defaults to 3), and each such process will use the amount of dedicated memory dictated by this parameter. This should be kept in mind when setting this parameter, especially on systems with relatively modest amounts of physical memory, so as to avoid swapping. Swapping has very negative effects on database performance and should be avoided. If the value recommended above is less than 256MB, it is chosen with this consideration in mind. However, the optimal value is workload-dependent, so it may be worthwhile to experiment with higher or lower settings.

|                |  |
|----------------|--|
| Rule           | Check <code>effective_io_concurrency</code>              |
| Recommendation | Consider adjusting <code>effective_io_concurrency</code> |
| Trigger        | <code>effective_io_concurrency &lt; 2</code>             |
| Severity       | Low  |

**Description:** If the PostgreSQL data files are located on a RAID array or SSD, `effective_io_concurrency` should be set to the approximate number of I/O requests that the system can service simultaneously. For RAID arrays, this is typically equal to the number of drives in the array. For SSDs, some experimentation may be needed to determine the most effective value. Setting this parameter to an appropriate value improves the performance of bitmap index scans. The default value of 1 is appropriate for cases where all PostgreSQL data files are located on a single spinning medium.

|                |  |
|----------------|--|
| Rule           | Check <code>fsync</code> is enabled            |
| Recommendation | Consider configuring <code>fsync = on</code> . |
| Trigger        | <code>fsync = off</code>                       |
| Severity       | High   |

**Description:** When `fsync` is set to off, a system crash can result in unrecoverable data loss or non-obvious corruption. `fsync = off` is an appropriate setting only if you are prepared to erase and recreate all of your databases in the event of a system crash or unexpected power outage.

Note: Much of the performance benefit obtained by configuring `fsync = off` can also be obtained by configuring `synchronous_commit = off`. However, the

latter settings is far safer: in the event of a crash, the last few transactions committed might be lost if they have not yet made it to disk, but the database will not be corrupted.

|                |   |
|----------------|---|
| Rule           | Check wal_sync_method   |
| Recommendation | On Windows, consider configuring wal_sync_method = fsync or wal_sync_method = fsync_writethrough. |
| Trigger        | OS = Windows and wal_sync_method not in ('fsync', 'fsync_writethrough')                           |
| Severity       | High  |

**Description:** In order to guarantee reliable crash recovery, PostgreSQL must ensure that the operating system flushes the write-ahead log to disk when asked to do so. On Windows, this can be achieved by setting wal\_sync\_method to fsync or fsync\_writethrough, or by disabling the disk cache on the drive where the write-ahead log is written. (It is safe to leave the disk cache enable if a battery-back disk cache is in use.)

Note: In cases where the loss of a very recently committed transaction is acceptable, the performance impact of flushing the write ahead log to disk can be mitigated by setting synchronous\_commit = off. In other situations, the use of a battery-backed RAID controller is recommended.

|                |   |
|----------------|---|
| Rule           | Check wal_sync_method   |
| Recommendation | On Mac OS X, consider configuring wal_sync_method = fsync_writethrough. |
| Trigger        | OS == MacOS X and wal_sync_method != fsync_writethrough                 |
| Severity       | High  |

**Description:** In order to guarantee reliable crash recovery, PostgreSQL must ensure that the operating system flushes the write-ahead log to disk when asked to do so. On MacOS X, this can be achieved by setting wal\_sync\_method to fsync\_writethrough or by disabling the disk cache on the drive where the write-ahead log is written. It is safe to leave the disk cache enable if a battery-back disk cache is in use.

Note: In cases where the loss of a very recently committed transaction is acceptable, the performance impact of flushing the write ahead log to disk can be mitigated by setting synchronous\_commit = off. In other situations, the use of a battery-backed RAID controller is recommended.

|                |   |
|----------------|---|
| Rule           | Check wal_buffers                       |
| Recommendation | Consider adjusting wal_buffers          |
| Trigger        | wal_buffers < 1MB or wal_buffers > 16MB |
| Severity       | Medium                                  |

**Description:** Increasing the configuration parameter wal\_buffers from the default value of 64kB to 1MB or more can reduced the number of times the database must flush the write-ahead log, leading to improved performance under some workloads. There is no benefit to setting this parameter to a value greater than the size of a WAL segment (16MB).

|                |                                    |
|----------------|------------------------------------|
| Rule           | Check commit_delay                 |
| Recommendation | Consider setting commit_delay = 0. |
| Trigger        | commit_delay != 0                  |
| Severity       | Low                                |

**Description:** Setting the commit\_delay configuration parameter to a non-zero value causes the system to wait for the specified number of microseconds before flushing the write-ahead log to disk at commit time, potentially allowing several concurrent transactions to commit with a single log flush. In most cases, this does not produce a performance benefit, and in some cases, it can produce a performance regression. Unless you have confirmed through benchmarking that a non-default value for this parameter produces a performance benefit, the default value of 0 is recommended.

|                |   |
|----------------|---|
| Rule           | Check checkpoint_segments                             |
| Recommendation | Consider adjusting checkpoint_segments.               |
| Trigger        | checkpoint_segments < 10 or checkpoint_segments > 300 |
| Severity       | Medium  |

**Description:** In order to ensure reliable and efficient crash recovery, PostgreSQL periodically writes all dirty buffers to disk. This process is called a checkpoint. Checkpoints occur when (1) the number of write-ahead log segments written since the last checkpoint exceeds checkpoint\_segments, (2) the amount of time since the last checkpoint exceeds checkpoint\_timeout, (3) the SQL command CHECKPOINT is issued, or (4) the system completes either shutdown or crash recovery. Increasing the value of checkpoint\_segments will reduce the frequency of checkpoints and will therefore improve performance, especially during bulk loading. The main downside of increasing checkpoint\_segments is that, in the event of a crash, recovery will require a longer period of time to return the database to a consistent state. In addition, increasing checkpoint\_segments will increase disk space consumption during periods of heavy system activity. However, because the theoretical limit on the amount of additional disk space that will be consumed for this reason is less than 32MB per additional checkpoint segment, this is often a small price to pay for improved performance.

Values between 30 and 100 are often suitable for modern systems. However, on smaller systems, a value as low as 10 may be appropriate, and on larger systems, a value as 300 may be useful. Values outside this range are generally not worthwhile.

|                |  |
|----------------|--|
| Rule           | Check checkpoint_completion_target               |
| Recommendation | Consider adjusting checkpoint_completion_target. |
| Trigger        | checkpoint_completion_target != 0.9              |
| Severity       | Medium   |

**Description:** In order to ensure reliable and efficient crash recovery, PostgreSQL periodically writes all dirty buffers to disk. This process is called a checkpoint. Beginning in PostgreSQL 8.3, checkpoints take place over an extended period of time in order to avoid swamping the I/O system. checkpoint\_completion\_target controls the rate at which the checkpoint is performed, as a function of the time remaining before the next checkpoint is due to start. A value of 0 indicates that the checkpoint should be performed as quickly as possible, whereas a value of 1 indicates that the checkpoint should complete just as the next checkpoint is scheduled to start. It is usually beneficial to spread the checkpoint out as much as possible; however, if checkpoint\_completion\_target is set to a value greater than 0.9, unexpected delays near the end of the checkpoint process can cause the checkpoint to fail to complete before the next one needs to start. Because of this, the recommended setting is 0.9.

|                   |  |
|-------------------|--|
| Rule              | Check effective_cache_size   |
| Recommendation    | Consider adjusting effective_cache_size.   |
| Trigger           | effective_cache_size < 0.5 * system_memory or effective_cache_size > MAX(0.9 * system_memory, system_memory - 1GB) |
| Recommended value | 0.75 * system_memory   |
| Severity          | Medium   |

**Description:** When estimating the cost of a nested loop with an inner index-scan, PostgreSQL uses this parameter to estimate the chances that rows from the inner relation which are fetched multiple times will still be in cache when the second fetch occurs. Changing this parameter does not allocate any memory, but an excessively small value may discourage the planner from using indexes that would in fact speed up the query. The recommended value is 75% of system memory.

|                   |   |
|-------------------|---|
| Rule              | Check default_statistics_target                                   |
| Recommendation    | Consider adjusting default_statistics_target.                     |
| Trigger           | default_statistics_target < 25 or default_statistics_target > 400 |
| Recommended value | 100   |
| Severity          | Medium  |

**Description:** PostgreSQL uses statistics to generate good query plans. These statistics are gathered either by a manual ANALYZE command or by an automatic analyze launched by the autovacuum daemon, and they include the most common values in each column of each database table, the approximate distribution of the remaining values, the fraction of rows which are NULL, and several other pieces of statistical information.

default\_statistics\_target indicates the level of detail that should be used in gathering and recording these statistics. A value of 100, which is the default beginning in PostgreSQL 8.4, is reasonable for most workloads. For very simple queries, a smaller value may be useful, while for complex queries especially against large tables, a higher value may work better. In some case, it can be helpful to override the default statistics target for specific table columns using ALTER TABLE .. ALTER COLUMN .. SET STATISTICS.

|                |   |
|----------------|---|
| Rule           | Check planner methods is enabled        |
| Recommendation | Avoid disabling planner methods.        |
| Trigger        | any <a href="#">enable</a> * GUC is off |
| Severity       | High                                    |

**Description:** The enable\_bitmapscan, enable\_hashagg, enable\_hashjoin, enable\_indexscan, enable\_material, enable\_mergejoin, enable\_nestloop, enable\_seqscan, enable\_sort, and enable\_tidscan parameters are intended primarily for debugging and should not be turned off. It can sometimes be helpful to disable one or more of these parameters for a particular query, when there is no other way to obtain the desired plan. However, none of these parameters should ever be turned off on a system-wide basis.

|                |   |
|----------------|---|
| Rule           | Check track_counts is enabled           |
| Recommendation | Consider configuring track_counts = on. |
| Trigger        | track_counts = off                      |
| Severity       | High                                    |

**Description:** Autovacuum will not function properly if track\_counts is disabled. Regular vacuuming is crucial to system stability and performance.

|                |                                       |
|----------------|---------------------------------------|
| Rule           | Check autovacuum is enabled           |
| Recommendation | Consider configuring autovacuum = on. |
| Trigger        | autovacuum = off                      |
| Severity       | High                                  |

**Description:** Enabling autovacuum is an important part of maintaining system stability and performance. Although disabling autovacuum may be useful during bulk loading, it should always be promptly reenabled when bulk loading is completed. Leaving autovacuum disabled for extended periods of time will result in table and index "bloat", where available free space is not reused, resulting in uncontrolled table and index growth. Reversing such bloat requires invasive maintenance using CLUSTER, REINDEX, and/or VACUUM FULL. Allowing autovacuum to work normally is usually sufficient to avoid the need for such maintenance.

|                |   |
|----------------|---|
| Rule           | Check configuring seq_page_cost                         |
| Recommendation | Consider configuring seq_page_cost <= random_page_cost. |
| Trigger        | seq_page_cost > random_page_cost                        |
| Severity       | Medium  |

**Description:** seq\_page\_cost and random\_page\_cost are parameters used by the query parameter to determine the optimal plan for each query. seq\_page\_cost represents the cost of a sequential page read, while random\_page\_cost represents the cost of a random page read. While these costs might be equal, if, for example, the database is fully cached in RAM, the sequential cost can never be higher. The PostgreSQL query planner will produce poor plans if seq\_page\_cost is set higher than random\_page\_cost.

|      |                                 |
|------|---------------------------------|
| Rule | Check reducing random_page_cost |
|------|---------------------------------|

|                |   |
|----------------|---|
| Recommendation | Consider reducing random_page_cost to no more than twice seq_page_cost. |
| Trigger        | random_page_cost > 2 * seq_page_cost                                    |
| Severity       | Low   |

**Description:** seq\_page\_cost and random\_page\_cost are parameters used by the query parameter to determine the optimal plan for each query. seq\_page\_cost represents the cost of a sequential page read, while random\_page\_cost represents the cost of a random page read. random\_page\_cost should always be greater than or equal to seq\_page\_cost, but it is rarely beneficial to set random\_page\_cost to a value more than twice seq\_page\_cost. However, the correct values for these variables are workload-dependent. If the database's working set is much larger than physical memory and the blocks needed to execute a query will rarely be in cache, setting random\_page\_cost to a value greater than twice seq\_page\_cost may maximize performance.

|                |  |
|----------------|--|
| Rule           | Check increasing seq_page_cost   |
| Recommendation | Consider increasing seq_page_cost.   |
| Trigger        | seq_page_cost < cpu_tuple_cost, seq_page_cost < cpu_index_tuple_cost, or seq_page_cost < cpu_operator_cost |
| Severity       | Medium   |

**Description:** The cost of reading a page into the buffer cache, even if it is already resident in the operating system buffer cache, is rarely less than the cost of a CPU operation. Thus, the value of the configuration parameter seq\_page\_cost should usually be greater than the values of the configuration parameters cpu\_tuple\_cost ,cpu\_index\_tuple\_cost, and cpu\_operator\_cost.

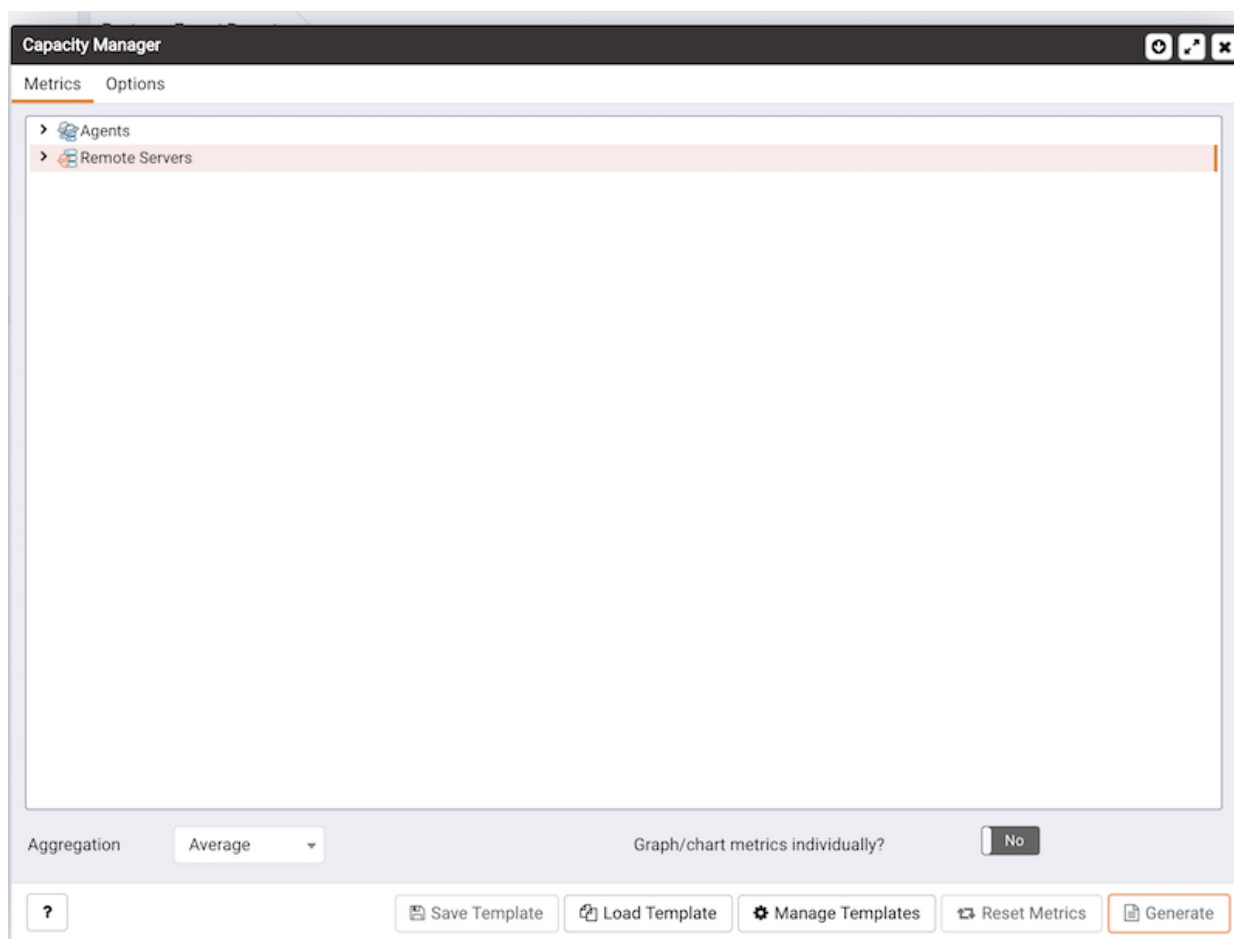
### 34.4.8 Capacity Manager

PEM's Capacity Manager analyzes collected statistics (metrics) to generate a graph or table that displays the historical usage statistics of an object, and can project the anticipated usage statistics for an object. You can configure Capacity Manager to collect and analyze metrics for a specific:

- Host/operating system
- EDB Postgres Advanced Server or PostgreSQL server
- Database
- Database object (table, index, function etc).

You can tailor the content of the Capacity Manager report by choosing a specific metric (or metrics) to include in the report, the time range over which the metrics were gathered, and a high or low threshold for the metrics analyzed. You can also specify a start and end date for the Capacity Manager report. If the end date of the report specifies a time in the future, Capacity Manager will analyze the **historical** usage of the selected object to extrapolate the **projected** object usage in the future.

To open Capacity Manager, select the **Capacity Manager...** option from the **Management** menu in the PEM client window; the **Capacity Manager** wizard opens, displaying a tree control on the **Metrics** tab.



Expand the tree control on the [Metrics](#) tab to select the metrics that will be included in the Capacity Manager report.

When defining report options, you can specify an [aggregation](#) method for each selected metric. The aggregation method determines how Capacity Manager will analyze the data points within the sampling period to reduce the data to a more visually meaningful quantity within a report (if required). The aggregation method can instruct Capacity Manager to compute an average of the data within a time period, the high or low value, or the first sampled value.

Use the [Options](#) tab to specify additional report details.

When defining the boundaries of a Capacity Manager report, specify the starting date and time, and an end boundary. The end boundary can be a point in time or a threshold boundary (when the data meets a specified criteria). If the sample contains more data points than the number of points specified by the [cm\\_data\\_points\\_per\\_report](#) configuration parameter, Capacity Manager applies the aggregation method to calculate a reduced number of graph points for the report.

## Report Templates

You can save a report definition as a template for future reports. Capacity Manager report templates may be accessed by all PEM users. To save a report definition as a template:

1. Use the [Metrics](#) and [Options](#) tabs to define your report.
2. Click the [Save](#) button to open the [Save Template](#) dialog.
3. Provide a report name in the Title field, select a location to store the template in the tree control.
4. Click [OK](#).

When creating a report, you can use the [Load Template](#) button to browse and open an existing template. Once opened, the report definition may be modified if required, and optionally saved again, either as a new template, or overwriting the original template. Use the [Manage Templates](#) button open a dialog that allows you to rename or remove unwanted templates.

## Available Metrics

Please Note that the available metrics will vary by platform, and are subject to change. The available metrics may include the metrics described in the table below:

| Metric Name                           | Description   |
|---------------------------------------|---|
| # Dead Tuples                         | The number of dead tuples in the selected table.  |
| # Dead Tuples+                        | The cumulative number of dead tuples in the selected table.   |
| # Heap Tuples Fetched by Index Scans  | The number of heap tuples fetched by index scans.   |
| # Heap Tuples Fetched by Index Scans+ | The cumulative number of heap tuples fetched by index scans.  |
| # Idle Backends+                      | The cumulative number of currently idle backend clients.  |
| # Index Scans                         | The number of index scans performed on the specified object.  |
| # Index Scans+                        | The cumulative number of index scans performed on the specified object.   |
| # Index Tuples Read                   | The number of index tuples read.  |
| # Index Tuples Read+                  | The cumulative number of index tuples read.   |
| # Live Tuples                         | The number of tuples visible to transactions.   |
| # Live Tuples+                        | The cumulative number of tuples visible to transactions.  |
| # Pages Estimated by ANALYZE          | The number of pages estimated by ANALYZE.   |
| # Pages Estimated by ANALYZE+         | The cumulative number of pages estimated by ANALYZE.  |
| # Sequential Scans                    | The number of sequential scans performed on the specific table.   |
| # Sequential Scans+                   | The cumulative number of sequential scans performed on the specific table.  |
| # Sequential Scan Tuples              | The number of tuples sequentially scanned in the specific table.  |
| # Sequential Scan Tuples+             | The cumulative number of tuples sequentially scanned in the specific table.   |
| # Tuples Deleted                      | The number of tuples deleted.   |
| # Tuples Deleted+                     | The cumulative number of tuples deleted.  |
| # Tuples Estimated by ANALYZE         | The number of live (visible) tuples estimated by ANALYZE.   |
| # Tuples Estimated by ANALYZE+        | The cumulative number of live tuples estimated by ANALYZE.  |
| # Tuples HOT Updated                  | The number of tuples HOT updated. In a HOT update, the new tuple resides in the same block as the original tuple and the tuples share an index entry. |
| # Tuples HOT Updated+                 | The cumulative number of tuples HOT updated.  |
| # Tuples Inserted                     | The number of tuples inserted into the specified table.   |
| # Tuples Inserted+                    | The cumulative number of tuples inserted into the specified table.  |
| # Tuples Updated                      | The number of tuples updated in the selected table.   |
| # Tuples Updated+                     | The cumulative number of tuples updated in the selected table.  |
| Blocks Hit                            | The number of blocks found in the cache.  |
| Blocks Hit+                           | The cumulative number of blocks found in the cache.   |
| Blocks Read                           | The number of blocks read.  |
| Blocks Read+                          | The cumulative number of blocks read.   |
| Blocks Read from InfiniteCache        | The number of blocks read from InfiniteCache.   |
| Blocks Read from InfiniteCache+       | The cumulative number of blocks read from InfiniteCache.  |
| Blocks Written                        | The number of blocks written.   |

| Metric Name                      | Description  |
|----------------------------------|--|
| Blocks Written+                  | The cumulative number of blocks written.   |
| Buffers Allocated                | The number of buffers allocated.   |
| Buffers Allocated+               | The cumulative number of buffers allocated.  |
| Buffers Written - Backends       | The number of buffer blocks written to disk by server processes (processes connected to a client application). |
| Buffers Written - Backends+      | The cumulative number of buffer blocks written to disk by server processes.                                    |
| Buffers Written - Checkpoint     | The number of blocks written to disk by the checkpoint process.  |
| Buffers Written - Checkpoint+    | The cumulative number of blocks written to disk by the checkpoint process.                                     |
| Buffers Written - Cleaning Scan  | The number of blocks written to disk by the autovacuum process.  |
| Buffers Written - Cleaning Scan+ | The cumulative number of blocks written to disk by the autovacuum process.                                     |
| Bytes Received (KB)              | The number of bytes received from the client (in kilobytes).   |
| Bytes Received (KB)+             | The cumulative number of bytes received (in kilobytes).  |
| Bytes Sent (KB)                  | The number of bytes sent to the client (in kilobytes).   |
| Bytes Sent (KB)+                 | The cumulative number of bytes sent (in kilobytes).  |
| Checkpoints - Timed              | The number of checkpoint operations triggered by the checkpoint interval.                                      |
| Checkpoints - Timed+             | The cumulative number of checkpoint operations triggered by the checkpoint interval.                           |
| Checkpoints - Untimed            | The number of checkpoint operations triggered by checkpoint size.  |
| Checkpoints - Untimed+           | The cumulative number of checkpoint operations triggered by checkpoint size.                                   |
| Database Size (MB)               | The size of the specified database (in megabytes).   |
| Free RAM Memory                  | The amount of free RAM memory (in megabytes).  |
| Free Swap Memory                 | The amount of free swap space on disk (in megabytes).  |
| Heap Blocks Hit                  | The number of heap blocks found in the cache.  |
| Heap Blocks Hit+                 | The cumulative number of heap blocks found in the cache.   |
| Heap Blocks Read                 | The number of heap blocks read.  |
| Heap Blocks Read+                | The cumulative number of heap blocks read.   |
| Index Blocks Hit                 | The number of index blocks found in the cache.   |
| Index Blocks Hit+                | The cumulative number of index blocks found in the cache.  |
| Index Blocks Read                | The number of index blocks read.   |
| Index Blocks Read+               | The cumulative number of index blocks read.  |
| Index Size (MB)                  | The size of the specified index (in megabytes).  |
| In Packets Discards              | The number of inbound packets discarded.   |
| In Packets Discards+             | The cumulative number of inbound packets discarded.  |
| In Packets Errors                | The number of inbound packets that contain errors.   |
| In Packets Errors+               | The cumulative number of inbound packets that contain errors.  |
| Link Bandwidth (Mbit/s)          | The speed of the network adapter (in megabits per second).   |
| Load Average - 15 Minute         | CPU saturation (in percent) - 15 minute sampling average.  |
| Load Average - 1 Minute          | CPU saturation (in percent) - 1 minute sampling average.   |
| Load Average - 5 Minute          | CPU saturation (in percent) - 5 minute sampling average.   |
| Load Percentage                  | CPU saturation in percent.   |
| Number of Prepared Transactions+ | The cumulative number of prepared transactions.  |
| Number of WAL Files+             | The cumulative number of write-ahead log files.  |
| Out Packets Discards             | The number of outbound packets discarded.  |
| Out Packets Discards+            | The cumulative number of outbound packets discarded.   |



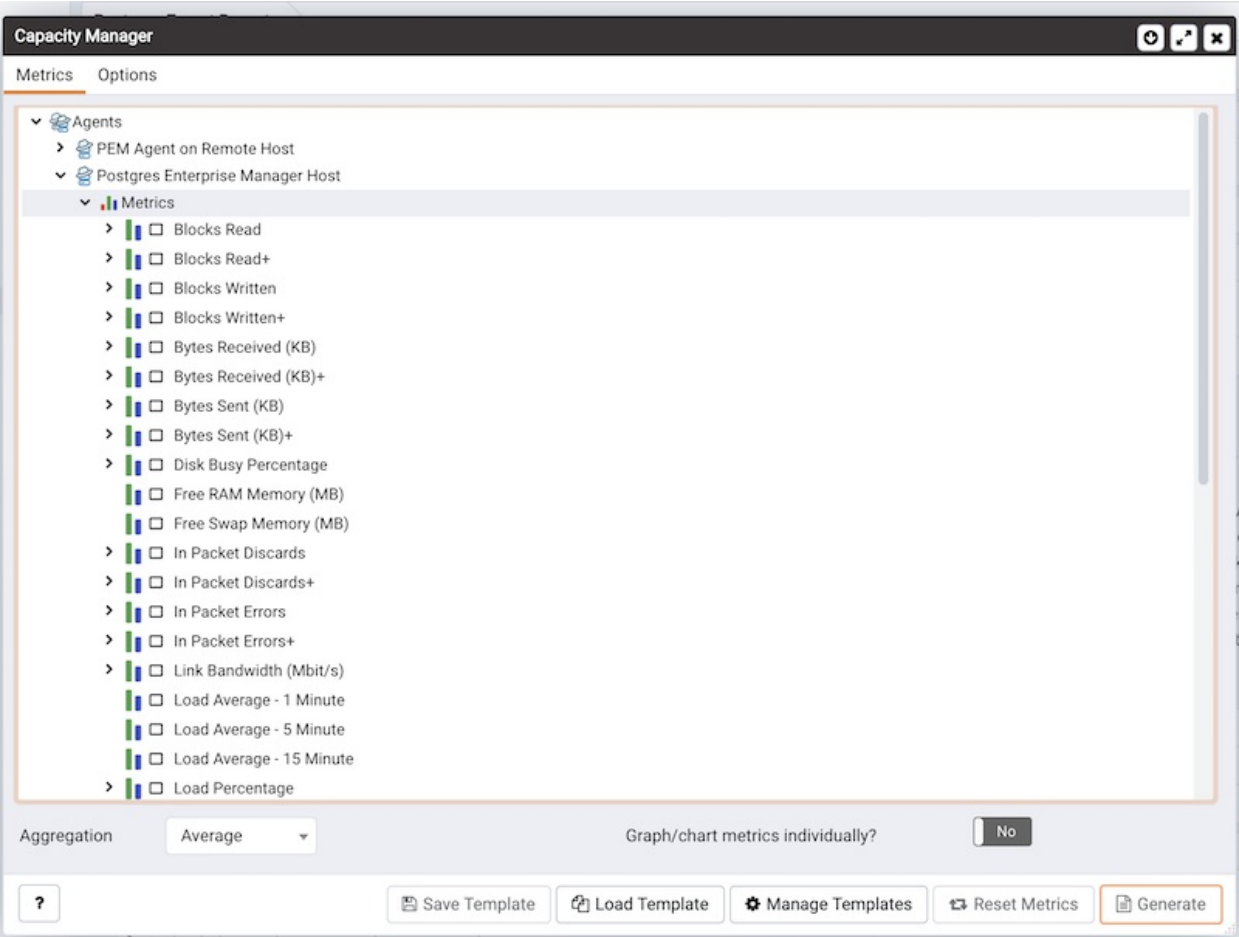
| Metric Name                              | Description  |
|--|--|
| Out Packets Errors                       | The number of outbound packets that contain errors.  |
| Out Packets Errors+                      | The cumulative number of outbound packets that contain errors.   |
| Packets Received                         | The number of packets received.  |
| Packets Received+                        | The cumulative number of packets received.   |
| Packets Sent                             | The number of packets sent.  |
| Packets Sent+                            | The cumulative number of packets sent.   |
| Size (MB)                                | The total size of the disk (in megabytes).   |
| Size of Indexes (MB)                     | The size of indexes on the specified table (in megabytes).   |
| Space Available (MB)                     | The current disk space available (in megabytes).   |
| Space Used (MB)                          | The current disk space used (in megabytes).  |
| Table Size (MB)                          | The size of the specified table (in megabytes).  |
| Tablespace Size (MB)                     | The size of the specified tablespace (in megabytes).   |
| Temp Buffers (MB)                        | The size of temporary buffers (in megabytes).  |
| Toast Blocks Hit                         | The number of TOAST blocks found in the cache.   |
| Toast Blocks Hit+                        | The cumulative number of TOAST blocks found in the cache.  |
| Toast Blocks Read                        | The number of TOAST blocks read.   |
| Toast Blocks Read+                       | The cumulative number of TOAST blocks read.  |
| Total RAM Memory                         | The total amount of RAM memory on the system (in megabytes).   |
| Total Swap Memory                        | The total amount of swap space on the system (in megabytes).   |
| Total Table Size w/Index<br>es and Toast | The total size of the specified table (including indexes and associated oversized attributes).   |
| Transactions Aborted                     | The number of aborted transactions.  |
| Transactions Aborted+                    | The cumulative number of aborted transactions.   |
| Transactions Committed                   | The number of committed transactions.  |
| Transactions Committed+                  | The cumulative number of committed transactions.   |
| Tuples Deleted                           | The number of tuples deleted from the specified table.   |
| Tuples Deleted+                          | The cumulative number of tuples deleted from the specified table.  |
| Tuples Estimated by<br>ANALYZE           | The number of visible tuples in the specified table.   |
| Tuples Estimated by<br>ANALYZE+          | The cumulative number of visible tuples in the specified table.  |
| Tuples Fetched                           | The number of tuples fetched from the specified table.   |
| Tuples Fetched+                          | The cumulative number of tuples fetched from the specified table.  |
| Tuples HOT Updated                       | The number of tuples HOT updated. In a HOT update, the new tuple resides in the same block as the original tuple and the tuples share an index entry.            |
| Tuples HOT Updated+                      | The cumulative number of tuples HOT updated. In a HOT update, the new tuple resides in the same block as the original tuple and the tuples share an index entry. |
| Tuples Inserted                          | The number of tuples inserted into the specified table.  |
| Tuples Inserted+                         | The cumulative number of tuples inserted into the specified table.   |
| Tuples Returned                          | The number of tuples returned in result sets.  |
| Tuples Returned+                         | The cumulative number of tuples returned in result sets.   |
| Tuples Updated                           | The number of tuples updated in the specified table.   |
| Tuples Updated+                          | The cumulative number of tuples updated in the specified table.  |
| WAL Segment Size (MB)                    | The segment size of the write-ahead log (in megabytes).  |

**Note:** The '+' following the name of a metric signifies that the data for the metric is gathered cumulatively; those metrics that are not followed by the '+' sign

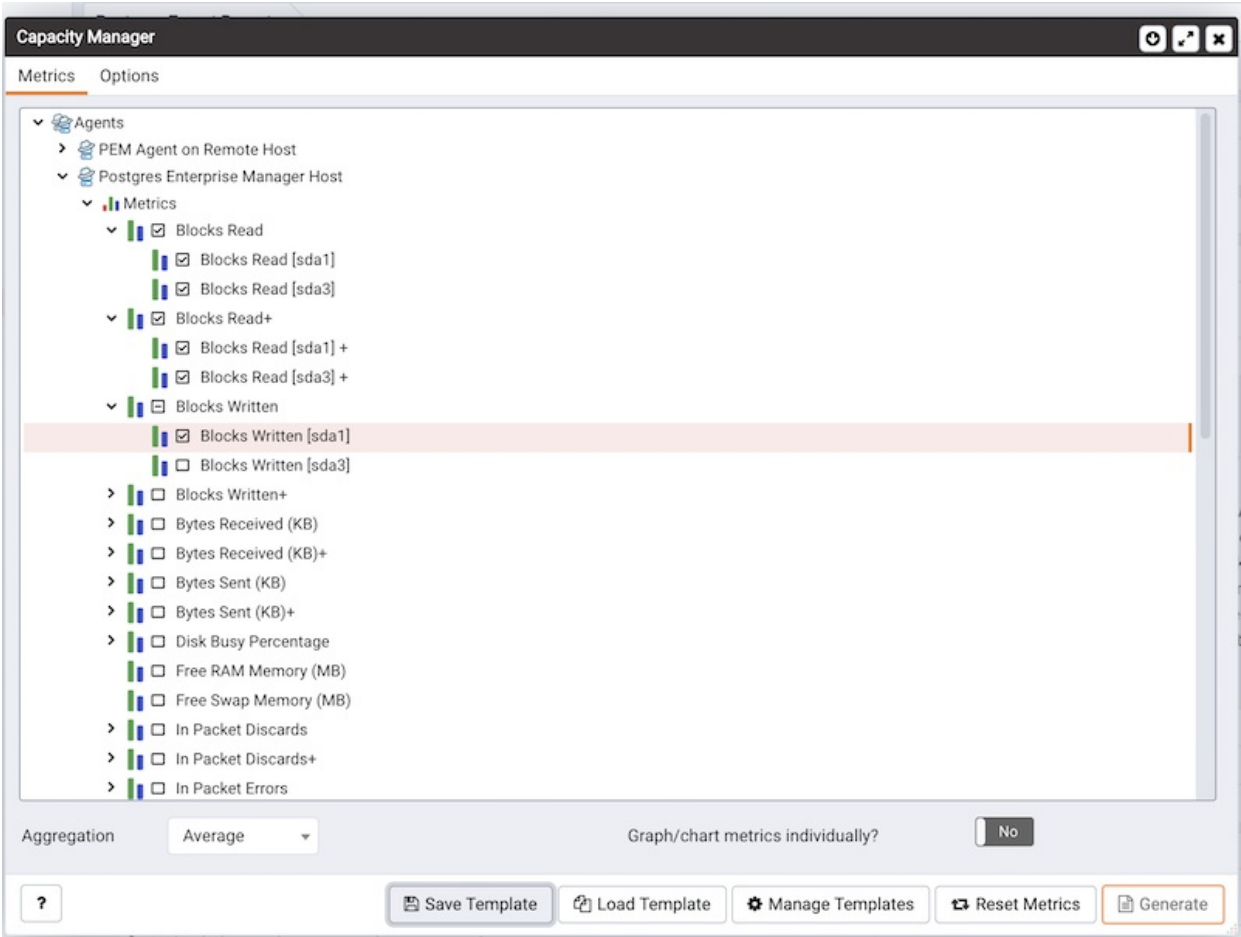
are collected as a 'point-in-time' value.

34.4.8.1 Tab 1 (Metrics)

To create a Capacity Manager report, expand the tree control on the Metrics tab to locate the metrics that are available for the node that you wish to analyze.



To include a metric in the Capacity Manager report, check the box to the left of the name of the metric on the Metrics tab.



Capacity Manager will use the aggregation method specified by the **Aggregation** drop-down listbox (located at the bottom of the **Metrics** tab). The aggregation method instructs Capacity Manager how to evaluate and plot the metric values. Select from:

- **Average:** Use the average of the values recorded during the time period.
- **Maximum:** Use the maximum value recorded during the time period.
- **Minimum:** Use the minimum value recorded during the time period.
- **First:** Use the first value recorded during the time period.

To remove a metric from the Capacity Manager report, uncheck the box to the left of the name of a metric.

Move the slider next to **Graph/chart metrics individually?** to **Yes** to instruct Capacity Manager to produce a separate report for each metric selected on the **Metrics** tab. If the option is set to **No** , all selected metrics will be merged into a single graph or table.

Click the **Generate** button to display the report onscreen (accepting the default configuration options), or continue to the **Options** tab to specify sampling boundaries, report type and report destination.

### 34.4.8.2 Tab 2 (Options)

Use the fields on the **Options** tab to specify the starting and ending boundaries of the Capacity Manager report, the type of report generated, and the location to which the report will be displayed or written.

Capacity Manager

MetricsOptions

Time Period

Period

Start time and end time

Start time

2020-04-23 17:56:01 +05:30

End time

2020-04-27 17:56:05 +05:30

Historical day(s)

7

Extrapolated day(s)

7

Threshold

Falls below

0

Report

Include on report

☒ Graph

☐ Table of data

Report destination

☒ New tab

☐ Previous tab

☐ Download the report as a file:

cm\_report.html

Note: Filename can only be latin-1 compatible characters.

?

Save TemplateLoad TemplateManage TemplatesReset MetricsGenerate

Use the fields within the **Time Period** box to define the boundaries of the Capacity Manager report:

- Use the **Period** drop-down listbox to select the type of time period you wish to use for the report. You can select:

| Value                                 | Description   |
|---------------------------------------|---|
| Start time and end time               | Specify a start and end date and time for the report.   |
| Start time and threshold              | Specify a start date and time, and a threshold to determine the end time and date for the report.   |
| Historical days and extrapolated days | Specify a start date for the report that is a number of days in the past, and an end date that is a number of days in the future. This option is useful for report templates that do not specify fixed dates. |
| Historical days and threshold         | Specify a start date that is a number of days in the past, and end it when a threshold value is reached.  |

After specifying the type of time period for the report, select from other options in the **Time Period** box to refine the time period:

- Use the date and time selectors next to the **Start time** field to specify the starting date and time of the sampling period, or select the number of **Historical day(s)** of data to include in the report. By default, Capacity Manager will select a start time that is one week prior to the current date and time. The date and time specified in the **Start time** field must not be later than the current date/time.
- Use the date and time selectors next to the **End time** field to specify an end boundary for the report, or select the number of **Extrapolated day(s)** of data to include in the report. The end boundary can be either a time, a number of days in the future, or the point at which a selected metric reaches a user-specified threshold value. The time specified in the **End time** field must be later than the time specified in the **Start time** field.

Note that if you select an end date and time in the future, Capacity Manager will use historical usage information to extrapolate anticipated future usage. Since the projected usage is based on the sampling of historical data, the accuracy of the future usage trend will improve with a longer sampling period.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

486

To specify a threshold value, use the drop-down listbox in the **Threshold** field to select a metric (from the metrics specified on the **Metrics** tab), an operator ( **Exceeds** or **Falls below** ), and to enter a target value for the metric. If you choose to define the end of the report using a threshold, the Capacity Manager report will terminate when the value for the selected metric exceeds or falls below the specified value.

**Please Note:** If you specify a starting boundary that is later than the ending boundary for the report, the status bar will display an error informing you that you must enter a valid time.

The `cm_max_end_date_in_years` configuration parameter defines a default time value for the end boundary of Capacity Manager reports. If you specify a threshold value as the end boundary of a report, and the anticipated usage of the boundary is not met before the maximum time has passed (as specified in the `cm_max_date_in_years` parameter), the report will terminate at the time specified by the `cm_max_date_in_years` parameter. By default, `cm_max_end_date_in_years` is 5; use the `Server Configuration` `dialog \<pem_server_config>` to modify the value of `cm_max_end_date_in_years`.

**Please Note:** The PEM client will display time in the PEM client's timezone, rather than the timezone in which the PEM server resides.

Use the fields in the **Report** box to specify the report type and destination.

The radio buttons next to **Include on report** specify the type of report produced by Capacity Manager. Choose from:

- Select **Graph** to instruct Capacity Manager to display the report in the form of a line graph in the PEM client window.
- Select **Table of data** to instruct Capacity Manager to display a table containing the report data in the PEM client window.
- Select **Graph and table of data** to instruct Capacity Manager to display both a line graph and a data table in the PEM client window.

Use the **Report destination** radio buttons to instruct Capacity Manager where to display or save the report:

- Select **New tab** to instruct Capacity Manager to display the report on a new tab in the PEM client. You must select **New tab** to display the first generation of a Capacity Manager report; for subsequent reports, you may select **Previous** tab.
- Select **Previous** tab to instruct Capacity Manager to re-use a previously opened tab when displaying the report.
- Select **Download the report as a file** and specify a file name to instruct Capacity Manager to write the report to the specified file.

Reports saved to file are stored in HTML format. You can review Capacity Manager reports with any web browser that supports Scalable Vector Graphics (SVG). Browsers that do not support SVG will be unable to display Capacity Manager graphs and may include unwanted characters.

When you have specified the report boundaries and selected the type and destination of the Capacity Manager report, click the **Generate** button to create the report.

## 34.4.9 Alerting

Postgres Enterprise Manager monitors a system for conditions that require user attention. An alert definition contains a system-defined or user-defined set of conditions that PEM compares to the system statistics; if the statistics deviate from the boundaries specified for that statistic, the alert triggers, displaying a **High** (red), **Low** (yellow) or **Medium** (orange) severity warning in the left-most column of the **Alerts Status** table on the **Global Overview** dashboard, and optionally sends a notification via email to **Email Groups** or **SNMP trap/notification receivers**.

| Alerts Status |            |                                    |  |                 |          |        |         |        |                     |
|---------------|------------|------------------------------------|--|-----------------|----------|--------|---------|--------|---------------------|
|               | Alarm Type | Object Description                 | Alert Name   | Value           | Database | Schema | Package | Object | Alerting Since      |
| ▶             | ● High     | EDB Postgres Advanced Server 11    | <a href="#">Database size in server</a>                | 113 MB          |          |        |         |        | 2020-04-22 11:50:00 |
| ▶             | ● High     | EDB Postgres Advanced Server 11    | <a href="#">Last Vacuum</a>                            | Never ran       |          |        |         |        | 2020-04-21 21:26:54 |
| ▶             | ● High     | EDB Postgres Advanced Server 11    | <a href="#">Last AutoVacuum</a>                        | 140.21 hrs      |          |        |         |        | 2020-04-22 12:04:05 |
| ▶             | ● High     | EPAS_12                            | <a href="#">Table size in server</a>                   | 410 MB          |          |        |         |        | 2020-04-09 15:53:51 |
| ▶             | ● Medium   | EPAS_12                            | <a href="#">Last Vacuum</a>                            | 5.18 hrs        |          |        |         |        | 2020-04-27 20:47:50 |
| ▶             | ● High     | EPAS_12                            | <a href="#">Database size in server</a>                | 455 MB          |          |        |         |        | 2020-04-09 15:52:50 |
| ▶             | ● Medium   | EPAS_12                            | <a href="#">Last AutoVacuum</a>                        | 5.16 hrs        |          |        |         |        | 2020-04-27 20:47:50 |
| ▶             | ● High     | N/A                                | <a href="#">Alert Errors</a>                           | 3               |          |        |         |        | 2020-01-21 14:26:04 |
| ▶             | ● High     | PGSQL12_Centos7_1                  | <a href="#">Server Down</a>                            | 1               |          |        |         |        | 2020-04-27 20:48:50 |
| ▶             | ● High     | PGSQL12_Centos7_1                  | <a href="#">Last Vacuum</a>                            | Never ran       |          |        |         |        | 2020-04-03 14:58:57 |
| ▶             | ● High     | PGSQL12_Centos7_1                  | <a href="#">Last AutoVacuum</a>                        | Never ran       |          |        |         |        | 2020-04-03 14:58:57 |
| ▶             | ● High     | Postgres Enterprise Manager Server | <a href="#">Largest index by table-size percentage</a> | 100 %           |          |        |         |        | 2020-04-21 22:07:52 |
| ▶             | ● High     | Postgres Enterprise Manager Server | <a href="#">Database size in server</a>                | 2.072265625 GB  |          |        |         |        | 2020-02-05 18:26:49 |
| ▶             | ● High     | Postgres Enterprise Manager Server | <a href="#">Table size in server</a>                   | 1.9814453125 GB |          |        |         |        | 2020-02-20 11:29:45 |
| ▶             | ● Medium   | Postgres Enterprise Manager Server | <a href="#">Connections in idle state</a>              | 12              |          |        |         |        | 2020-04-27 16:20:32 |
| ▶             | ● Medium   | Postgres Enterprise Manager Server | <a href="#">Last Vacuum</a>                            | 4.99 hrs        |          |        |         |        | 2020-04-27 20:47:50 |

The PEM server includes a number of pre-defined alerts that are actively monitoring your servers. If the alert definition makes details available about the cause of the alert, you can click the down arrow to the right of the severity warning to access a dialog with detailed information about the condition that triggered the alert. Please note that Alert Details section lists top 10 entries only in the general tab.

Alert Details (Auto-refresh paused whilst rows are expanded. ⓘ)

|   | Ack'd                    | Alert Type | Name                                 | Value           | Agent | Server                             | Database | Schema | Package | Object | Alerting Since      |
|---|--------------------------|------------|--------------------------------------|-----------------|-------|------------------------------------|----------|--------|---------|--------|---------------------|
| ▶ | <input type="checkbox"/> | ● High     | <a href="#">Table size in server</a> | 1.9814453125 GB |       | Postgres Enterprise Manager Server |          |        |         |        | 2020-02-20 11:29:45 |

General

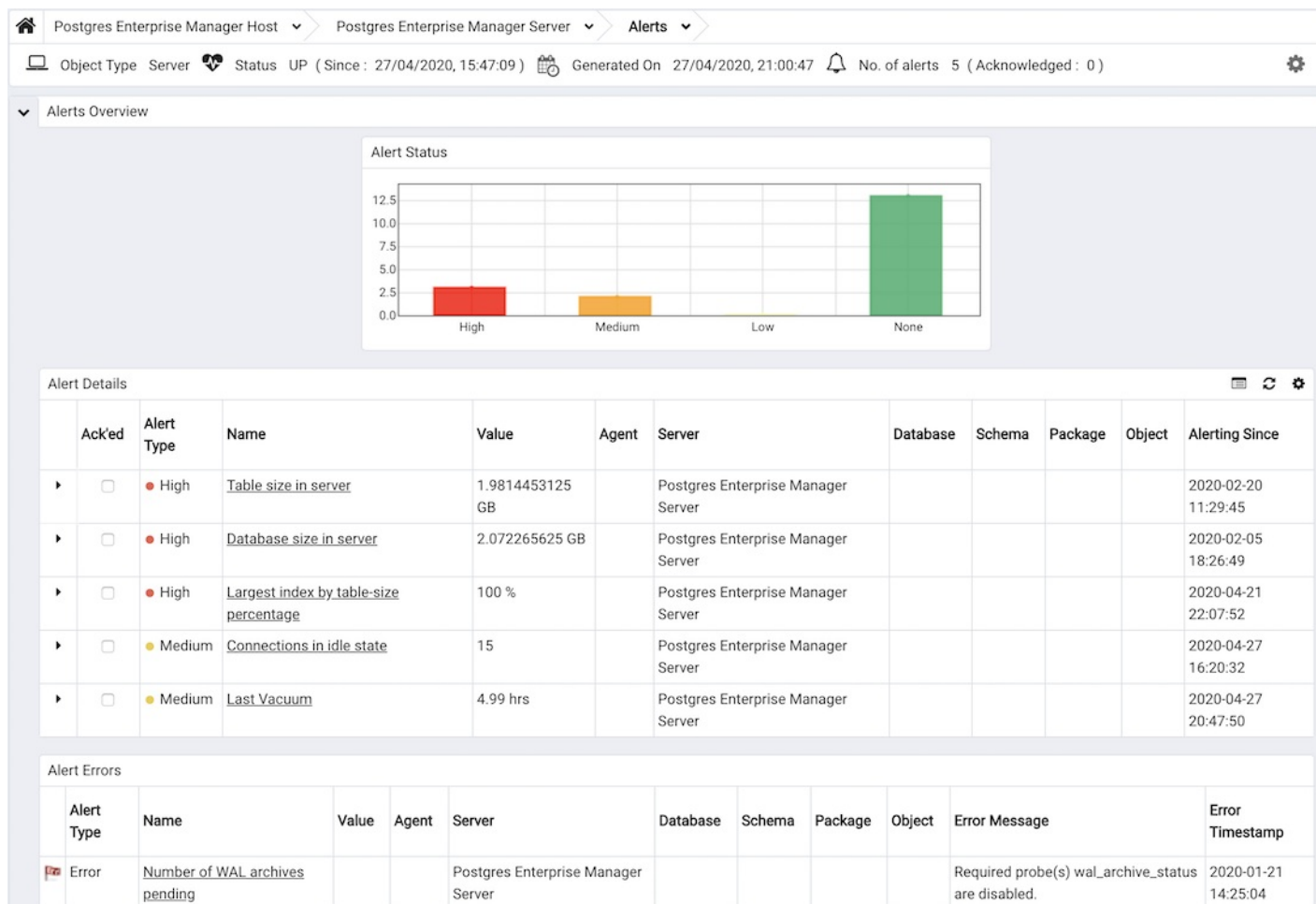
Parameters

| Table name       | Schema name | Database name | Total table size(MB) |
|------------------|-------------|---------------|----------------------|
| table_statistics | pemhistory  | pem           | 1087                 |
| server_logs      | pemdata     | pem           | 263                  |
| index_statistics | pemhistory  | pem           | 237                  |
| session_info     | pemhistory  | pem           | 137                  |
| lock_info        | pemhistory  | pem           | 88                   |

PEM also provides an interface that allows you to create customized alerts. Each alert uses metrics defined on an alert template. An alert template defines how the server will evaluate the statistics for a resource or metric. The PEM server includes a number of pre-defined alert templates, or you can create custom alert templates.

Using the Alerts Dashboard

Use the **Dashboards** menu (at the top of the **Global Overview** dashboard) to access the **Alerts Dashboard**. The Alerts Dashboard displays a summary of the active alerts and the status of each alert:



The **Alerts Overview** section displays a graphic representation of the active alerts, as well as a count of the current High, Low and Medium alerts. The vertical bar on the left of the graph provides the count of the alerts displayed in each column. Hover over a bar to display the alert count for the selected alert severity in the upper-right hand corner of the **Alerts Status** graph.

The **Alert Details** table provides a list of the alerts that are currently triggered. The entries are prioritized from high-severity to lower-severity; each entry includes information that will allow you to identify the alert and recognize the condition that triggered the alert. Click the name of an alert to review the alert definition, or the down arrow next to the alert icon to review the metrics that triggered the alert.

The **Alert Errors** table displays configuration-related errors (eg. accidentally disabling a required probe, or improperly configuring an alert parameter). You can use the information provided in the **Error Message** column to identify and resolve the conflict that is causing the error; for additional assistance, contact [EnterpriseDB Support](#).

## Managing Alerts

PEM's **Manage Alerts** tab allows you to define custom alerts or modify existing alerts. To open the **Manage Alerts** tab, select **Manage Alerts...** from the **Management** menu. The Manage Alerts tab provides an easy way to review the alerts that are currently defined for the object that is highlighted in the PEM client tree control; simply select an object to see the alerts that are defined for that object.



Quick Links

Copy Alerts

Alert Templates

Email Groups

Webhooks

Server Configuration

Help

Alerts

Manage Alerts

+

🔍

🔄

|                                     | Name  | Auto created?  | Template  | Enable?        | Interval       |         | History retention |      |
|-------------------------------------|---|----------------|---|----------------|----------------|---------|-------------------|------|
|                                     |   |                |   |                | Default?       | Minutes | Default?          | Days |
| <input checked="" type="checkbox"/> | <div>Average table bloat in server</div>                  | <div>Yes</div> | <div>Average table bloat in server</div>        | <div>Yes</div> | <div>Yes</div> | 1       | <div>Yes</div>    | 30   |
| <input checked="" type="checkbox"/> | <div>Connections in idle-in-transaction state</div>       | <div>Yes</div> | <div>Connections in idle-in-transactio...</div> | <div>Yes</div> | <div>Yes</div> | 1       | <div>Yes</div>    | 30   |
| <input checked="" type="checkbox"/> | <div>Connections in idle-in-transaction state, a...</div> | <div>Yes</div> | <div>Connections in idle-in-transactio...</div> | <div>Yes</div> | <div>Yes</div> | 1       | <div>Yes</div>    | 30   |
| <input checked="" type="checkbox"/> | <div>Connections in idle state</div>                      | <div>Yes</div> | <div>Connections in idle state</div>            | <div>Yes</div> | <div>Yes</div> | 1       | <div>Yes</div>    | 30   |
| <input checked="" type="checkbox"/> | <div>Database size in server</div>                        | <div>Yes</div> | <div>Database size in server</div>              | <div>Yes</div> | <div>Yes</div> | 1       | <div>Yes</div>    | 30   |
| <input checked="" type="checkbox"/> | <div>Highest table bloat in server</div>                  | <div>Yes</div> | <div>Highest table bloat in server</div>        | <div>Yes</div> | <div>Yes</div> | 1       | <div>Yes</div>    | 30   |
| <input checked="" type="checkbox"/> | <div>Largest index by table-size percentage</div>         | <div>Yes</div> | <div>Largest index by table-size perce...</div> | <div>Yes</div> | <div>Yes</div> | 1       | <div>Yes</div>    | 30   |
| <input checked="" type="checkbox"/> | <div>Last AutoVacuum</div>                                | <div>Yes</div> | <div>Last AutoVacuum</div>                      | <div>Yes</div> | <div>Yes</div> | 1       | <div>Yes</div>    | 30   |
| <input checked="" type="checkbox"/> | <div>Last Vacuum</div>                                    | <div>Yes</div> | <div>Last Vacuum</div>                          | <div>Yes</div> | <div>Yes</div> | 1       | <div>Yes</div>    | 30   |

The **Manage Alerts** tab also provides **Quick Links** that provide quick access to dialogs that allow you to:

- [Copy an alert](#) from one object to one or more objects.
- [Create or modify an alert template](#).
- [Create or Modify an email group](#).
- Manage [PEM Server configuration](#) details.
- Access the PEM online help.

You can configure an alert to notify Nagios network-alerting software when that alert is triggered. For more information, see [Using PEM with Nagios](#).


To [create a new alert](#), click the add icon in the upper-right corner of the **Alerts** table.


### 34.4.9.1 Creating and Managing Alerts


Use options accessed through the **Manage Alerts** tab to create, copy, or modify an alert. To open the **Manage Alerts** tab, select **Manage Alerts...** from the PEM client's **Management** menu.





Quick Links


Copy Alerts

Alert Templates

Email Groups










Webhooks

Server Configuration

Help

Alerts

Manage Alerts

|                                     |   | Name   | Auto created?                           | Template                             | Enable?                                 | Interval                                |         | History retention                       |      |  |
|-------------------------------------|---|--|---|--------------------------------------|---|---|---------|---|------|--|
|                                     |   |  |   |                                      |   | Default?                                | Minutes | Default?                                | Days |  |
| <input checked="" type="checkbox"/> |  | Average table bloat in server                  | <input checked="" type="checkbox"/> Yes | Average table bloat in server        | <input checked="" type="checkbox"/> Yes | <input checked="" type="checkbox"/> Yes | 1       | <input checked="" type="checkbox"/> Yes | 30   |  |
| <input checked="" type="checkbox"/> |  | Connections in idle-in-transaction state       | <input checked="" type="checkbox"/> Yes | Connections in idle-in-transactio... | <input checked="" type="checkbox"/> Yes | <input checked="" type="checkbox"/> Yes | 1       | <input checked="" type="checkbox"/> Yes | 30   |  |
| <input checked="" type="checkbox"/> |  | Connections in idle-in-transaction state, a... | <input checked="" type="checkbox"/> Yes | Connections in idle-in-transactio... | <input checked="" type="checkbox"/> Yes | <input checked="" type="checkbox"/> Yes | 1       | <input checked="" type="checkbox"/> Yes | 30   |  |
| <input checked="" type="checkbox"/> |  | Connections in idle state                      | <input checked="" type="checkbox"/> Yes | Connections in idle state            | <input checked="" type="checkbox"/> Yes | <input checked="" type="checkbox"/> Yes | 1       | <input checked="" type="checkbox"/> Yes | 30   |  |
| <input checked="" type="checkbox"/> |  | Database size in server                        | <input checked="" type="checkbox"/> Yes | Database size in server              | <input checked="" type="checkbox"/> Yes | <input checked="" type="checkbox"/> Yes | 1       | <input checked="" type="checkbox"/> Yes | 30   |  |
| <input checked="" type="checkbox"/> |  | Highest table bloat in server                  | <input checked="" type="checkbox"/> Yes | Highest table bloat in server        | <input checked="" type="checkbox"/> Yes | <input checked="" type="checkbox"/> Yes | 1       | <input checked="" type="checkbox"/> Yes | 30   |  |
| <input checked="" type="checkbox"/> |  | Largest index by table-size percentage         | <input checked="" type="checkbox"/> Yes | Largest index by table-size perce... | <input checked="" type="checkbox"/> Yes | <input checked="" type="checkbox"/> Yes | 1       | <input checked="" type="checkbox"/> Yes | 30   |  |
| <input checked="" type="checkbox"/> |  | Last AutoVacuum                                | <input checked="" type="checkbox"/> Yes | Last AutoVacuum                      | <input checked="" type="checkbox"/> Yes | <input checked="" type="checkbox"/> Yes | 1       | <input checked="" type="checkbox"/> Yes | 30   |  |
| <input checked="" type="checkbox"/> |  | Last Vacuum                                    | <input checked="" type="checkbox"/> Yes | Last Vacuum                          | <input checked="" type="checkbox"/> Yes | <input checked="" type="checkbox"/> Yes | 1       | <input checked="" type="checkbox"/> Yes | 30   |  |

Use the **Quick Links** toolbar to open dialogs and tabs that you can use to manage alerts and alerting behavior:


- Select **Copy Alerts** to open the **Copy Alert Configuration** dialog and copy an alert definition.
- Select **Alert Templates** to open the **Alert Template** tab, and create or modify an alert template.
- Select **Email Groups** to open the **Email Groups** tab, and manage or create an email group.
- Select **Webhooks** to open the **Webhooks** tab, and manage or create a webhook endpoint.
- Select **Server Configuration** to open the **server configuration** dialog and review or modify server configuration settings.
- Select **Help** to open the PEM online help.

The **Alerts** table displays the alerts that are defined for the item currently highlighted in the PEM client tree control. You can use the **Alerts** table to modify an existing alert, or to create a new alert.

## Creating a New Alert

To open the alert definition dialog and create a new alert, click the **Add** icon (+) in the upper-right corner of the table.

General Notification Script execution

**Name** 

**Description**

**Template**

A template uses metrics to generate a value to which PEM compares user specified alert boundaries. If the value returned by the template function evaluates to a value that is within the boundary of a user defined alert, PEM raises an alert.

**Enable?** ☒ Yes ☐ No  
Select Yes to enable the alert, and No to disable the alert.

**Interval**

**Default?** ☒ Yes ☐ No **Minutes**

Use fields in the Interval box to specify how often the alert should confirm that alert conditions are satisfied.

**History retention**

**Default?** ☒ Yes ☐ No **Days**

Use fields in the History retention box to specify the number of days that PEM will store data collected by the alert.

**Threshold values**

**Operator**  **Low**  **Medium**  **High**

**Unit**

The fields in the Threshold values box work together to define the triggering criteria for the alert.

**Auto created?** ☐ No ☒ Yes

**Parameter Options**

| Name   | Value                              |
|--------|------------------------------------|
| server | Postgres Enterprise Manager Server |

Use the fields on the **General** tab to provide information about the alert:

- Enter the name of the alert in the **Name** field.
- Use the drop-down listbox in the **Template** field to select a template for the alert. An alert template is a function that uses one (or more) metrics or parameters to generate a value to which PEM compares user-specified alert boundaries. If the value returned by the template function evaluates to a value that is within the boundary of a user-defined alert (as specified by the **Operator** and **Threshold values** fields), PEM raises an alert, adds a notice to the **Alerts overview** display, and performs any actions specified on the template.
- Use the **Enable?** switch to specify if the alert is enabled (**Yes**) or disabled (**No**).
- Use the controls in the **Interval** box to specify how often the alert should confirm if the alert conditions are satisfied. Use the **Minutes** selector to specify an interval value. Use the **Default** switch to set or reset the **Minutes** value to the default (recommended) value for the selected template.
- Use controls in the **History retention** box to specify the number of days that PEM will store data collected by the alert. Use the **Days** selector to specify the number of days that the data will be stored. Use the **Default** switch to set or reset the **Days** value to the default value (30 days).
- Use controls in the **Threshold values** box to define the triggering criteria for the alert. When the value specified in the **Threshold Values** fields evaluates to greater-than or less-than the system value (as specified with the **Operator**), PEM will raise a **Low**, **Medium** or **High** level alert:
- Use the **Operator** drop-down listbox to select the operator that PEM will use when evaluating the current system values.
  - Select a greater-than sign (>) to indicate that the alert should be triggered when the system values are greater than the values entered in the **Threshold values** fields.
  - Select a less-than sign (<) to indicate that the alert should be triggered when the system values are less than the values entered in the **Threshold values** fields.

- Use the threshold fields to specify the values that PEM will compare to the system values to determine if an alert should be raised. Please note that you must specify values for all three thresholds ( **Low** , **Medium** , and **High** ):
  - Enter a value that will trigger a low-severity alert in the **Low** field.
  - Enter a value that will trigger a medium-severity alert in the **Medium** field.
  - Enter a value that will trigger a high-severity alert in the **High** field.
- The **Unit** field takes the value as per the threshold unit defined in the selected alert template. You cannot add or edit the value in the **Unit** field while creating or editing the alert. The value can only be added or edited in the custom alert template.

The **Parameter Options** table contains a list of parameters that are required by the selected template; the table displays both pre-defined parameters, and parameters for which you must specify a value. Please note that you must specify a value for any parameter that displays a prompt in the **Value** column.

Use the **Notification** tab to specify how PEM will behave if an alert is raised.

GeneralNotificationScript execution

EmailWebhookSNMPNagios

To configure notification for an alert use the below fields to specify one or multiple user group that will receive an email notification if the alert is triggered at the specific level. Use the dropdown list box to select pre-defined group that will be sent an alert of the selected level is triggered. Please note that you must configure the PEM Server to use an SMTP server to deliver email before PEM can send email notifications.

Alert types

All alerts?

No

<Default>

Low alerts?

No

<Default>

Medium alerts?

No

<Default>

High alerts?

No

<Default>

PEM can send a notification or execute a script if an alert is triggered, or if an alert is cleared.

Use the fields in the **Email** tab to specify the email group that will receive an email notification if the alert is triggered at the specified level. Use the **Email Groups** tab to create an email group that contains the address of the user or users that will be notified when an alert is triggered. To access the **Email Groups** tab, click the **Email Groups** icon located in the **Quick Links** menu of the **Manage Alerts** tab.

To instruct PEM to send an email when a specific alert level is reached, set the slider next to an alert level to **Yes** , and use the drop-down listbox to select the pre-defined user or group that will be notified.

Please note that you must [configure the PEM Server](#) to use an SMTP server to deliver email before PEM can send email notifications.

General Notification Script execution

Email Webhook SNMP Nagios

To configure notification for an alert use the below fields to specify one or multiple endpoints if the alert is triggered at the specified level. Use the dropdown to select a pre-defined endpoints that will be sent a notification.

Enable? ☒ Yes

Override default configuration? ☐ No

Alert types

Low alerts?

Medium alerts?

High alerts?

Cleared alerts?

Use the fields in the **Webhook** tab to specify the webhook endpoints that will receive a notification if the alert is triggered at the specified level. Use the **Webhooks** tab to create an endpoint that contains the details of URL that will be notified when an alert is triggered along with other details like payload. To access the **Webhooks** tab, click the **Webhooks** icon located in **Quick Links** menu of the **Manage Alerts** tab.

By default **Webhook** notifications will be sent to created endpoints according to their default settings. To disable the **Webhook** set the slider next to **Enable** field to **No**.

Also to override default settings set the slider next to **Override default configuration?** to **Yes**, and use the drop-down listbox to select the pre-defined endpoints.

General Notification Script execution

Email Webhook SNMP Nagios

Use the trap notification options to configure trap notifications for this alert Note that you must configure PEM server to send notifications to an SNMP trap/notification receiver before notifications can be sent.

Trap notification

Send trap? ☐ No

SNMP version

Low alert? ☐ No

Medium alert? ☐ No

High alert? ☐ No

Use the **Trap notification** options to configure trap notifications for this alert:

- Set the **Send trap** slider to **Yes** to send SNMP trap notifications when the state of this alert changes.
- Set the **SNMP Ver** to **v1**, **v2**, or **v3** to identify the SNMP version.
- Use the **Low alert**, **Med alert** and **High alert** sliders to select the level(s) of alert that will trigger the trap. For example, if you set the slider next to **High alert** to **Yes**, PEM will send a notification when an alert with a high severity level is triggered.

Please note that you must **configure the PEM Server** to send notifications to an SNMP trap/notification receiver before notifications can be sent. For sending SNMP v3 traps, pemAgent will use 'User Security Model(USM)' which is in charge of authenticating, encrypting, and decrypting SNMP packets.

Also note while sending SNMP v3 traps, agent will create snmp\_boot\_counter file. This file will get created in location mentioned by batch\_script\_dir parameter in agent.cfg, if this parameter is not configured or if directory is not accessible due to authentication restrictions then in operating systems

temporary directory, if that is also not possible then in user's home directory.

Please see [How SNMP traps are formed?](#)

The screenshot shows the 'Notification' tab in the configuration interface. Under the 'Nagios' sub-tab, there is a section titled 'Nagios notification'. It contains a label 'Submit passive service check result to Nagios?' followed by a toggle switch currently set to 'No'. Above this section, a note states: 'Set "Submit passive service check result to Nagios" to "Yes" to instruct the PEM server to notify Nagios when the alert is triggered or cleared.'

Use the field in the **Nagios notification** box to instruct the PEM server to notify Nagios network-alerting software when the alert is triggered or cleared. For detailed information about configuring and using Nagios with PEM, please see [Using PEM with Nagios](#).

- Set the **Submit passive service check result to Nagios** switch to **Yes** to instruct the PEM server to notify Nagios when the alert is triggered or cleared.

The screenshot shows the 'Script execution' tab in the configuration interface. It contains several settings:
 

- 'Execute script?' with a toggle switch set to 'No'.
- 'Execute on alert cleared?' with a toggle switch set to 'No'.
- 'Execute script on' with radio buttons for 'PEM Server' and 'Monitored Server', where 'Monitored Server' is selected.
- 'Code' with a large text area for entering script code.

Use the fields in the Script execution box to (optionally) define a script that will be executed if an alert is triggered, and to specify details about the script execution.

- Set the Execute script slider to Yes to instruct PEM to execute the provided script if an alert is triggered.
- Set the Execute on alert cleared slider to Yes to instruct PEM to execute the provided script when the situation that triggered the alert has been resolved.
- Use the selector to indicate if the script should execute on the PEM Server or the Monitored Server.
- Provide the script that PEM should execute in the Code field. You can provide a batch/shell script, or SQL code. Within the script you can use the placeholders to replace the following:
  - %AlertID% - the id of the triggered alert.
  - %AlertName% - the name of the triggered alert.
  - %ObjectName% - the name of the server or agent on which the alert was triggered.
  - %ObjectType% - the type on which alert was generated.
  - %ThresholdValue% - the threshold value reached by the metric when the alert triggered.
  - %CurrentValue% - the current value of the metric that triggered the alert.
  - %CurrentState% - the current state of the alert.
  - %OldState% - the previous state of the alert.
  - %AlertRaisedTime% - the time that the alert was raised, or the most recent time that the alert state was changed.
  - %AgentID% - the id of the agent by which alert was generated.
  - %AgentName% - the name of the agent by which alert was generated.
  - %ServerID% - the id of the server on which alert was generated.
  - %ServerName% - the name of the server on which alert was generated.
  - %ServerIP% - the ip or address of the server on which alert was generated.
  - %ServerPort% - the the port of the server on which alert was generated.
  - %DatabaseName% - the name of the database on which alert was generated.
  - %SchemaName% - the name of the schema on which alert was generated.
  - %PackageName% - the name of the package on which alert was generated.
  - %DatabaseObjectName% - the name of the database object name like table name, function name etc on which alert was generated.
  - %Parameters% - the list of custom parameters used to generate the alert.

Use the fields in the **Script execution** tab to (optionally) define a script that will be executed if an alert is triggered, and to specify details about the script execution.

- Set the **Execute script** slider to **Yes** to instruct PEM to execute the provided script if an alert is triggered.
- Set the **Execute on alert cleared** slider to **Yes** to instruct PEM to execute the provided script when the situation that triggered the alert has been resolved.
- Use the radio buttons next to **Execute script on** to indicate that the script should execute on the **PEM Server** or the **Monitored Server**.
- Provide the script that PEM should execute in the **Code** field. You can provide a batch/shell script, or SQL code. Within the script, you can use placeholders for the following:
  - **%AlertID%** - this placeholder will be replaced with the id of the triggered alert.
  - **%AlertName%** - this placeholder will be replaced with the name of the triggered alert.
  - **%ObjectName%** - this placeholder will be replaced with the name of the server or agent on which the alert was triggered.
  - **%ObjectType%** - this placeholder will be replaced with the type of the object on which the alert was triggered.
  - **%ThresholdValue%** - this placeholder will be replaced with the threshold value reached by the metric when the alert triggered.
  - **%CurrentValue%** - this placeholder will be replaced with the current value of the metric that triggered the alert.
  - **%CurrentState%** - this placeholder will be replaced with the current state of the alert.
  - **%OldState%** - this placeholder will be replaced with the previous state of the alert.
  - **%AlertRaisedTime%** - this placeholder will be replaced with the time that the alert was raised, or the most recent time that the alert state was changed.
  - **%AgentID%** - this placeholder will be replaced with the id of the agent by which alert was generated.
  - **%AgentName%** - this placeholder will be replaced with the name of the agent by which alert was generated.
  - **%ServerID%** - this placeholder will be replaced with the id of the server on which alert was generated.
  - **%ServerName%** - this placeholder will be replaced with the name of the server on which alert was generated.
  - **%ServerIP%** - this placeholder will be replaced with the IP or address of the server on which alert was generated.
  - **%ServerPort%** - this placeholder will be replaced with the port of the server on which alert was generated.
  - **%DatabaseName%** - this placeholder will be replaced with the name of the database on which alert was generated.
  - **%SchemaName%** - this placeholder will be replaced with the name of the schema on which alert was generated.
  - **%PackageName%** - this placeholder will be replaced with the name of the package on which alert was generated.
  - **%DatabaseObjectName%** - this placeholder will be replaced with the name of the database object on which alert was generated.
  - **%Parameters%** - this placeholder will be replaced with the list of custom parameters used to generate the alert.
  - **%AlertInfo%** - this placeholder will be replaced with the detailed database object level information of the alert.

When you have defined the alert attributes, click the edit icon to close the alert definition editor, and then the save icon (in the upper-right corner of the **Alerts** table). To discard your changes, click the refresh icon; a popup will ask you to confirm that you wish to discard the changes.

#### Note

Suppose you need to use the alert configuration placeholder values in an external script. You can do it either by passing them as the command-line arguments or exporting them as environment variables. Please note that the external script must have proper execution permissions.

- You can run the script with any of the placeholders as command-line argument.

For eg:

```
#!/bin/bash

bash <path_to_script>/script.sh "%AlertName% %AlertLevel% %AlertDetails%"
```

- You can define the environment variables for any of the placeholders and then use those environment variables in the script.

For eg:

```
#!/bin/bash

export AlertName=%AlertName%
export AlertState=%AlertState%

bash <path_to_script>/script.sh
```



## Modifying an Existing Alert

Use the **Alerts** table to manage an existing alert or create a new alert. Highlight an object in the PEM client tree control to view the alerts that monitor that object.

| Alerts                              |  |                                     |                                      |                                     |                                     |         |                                     |      |
|-------------------------------------|--|-------------------------------------|--------------------------------------|-------------------------------------|-------------------------------------|---------|-------------------------------------|------|
|                                     | Name   | Auto created?                       | Template                             | Enable?                             | Interval                            |         | History retention                   |      |
|                                     |  |                                     |                                      |                                     | Default?                            | Minutes | Default?                            | Days |
| <input checked="" type="checkbox"/> | Audit config mismatch                              | <input checked="" type="checkbox"/> | Audit config mismatch                | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30   |
| <input checked="" type="checkbox"/> | Average table bloat in server                      | <input checked="" type="checkbox"/> | Average table bloat in server        | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30   |
| <input checked="" type="checkbox"/> | Connections in idle-in-transaction state           | <input checked="" type="checkbox"/> | Connections in idle-in-transactio... | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30   |
| <input checked="" type="checkbox"/> | Connections in idle-in-transaction state, as a ... | <input checked="" type="checkbox"/> | Connections in idle-in-transactio... | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30   |
| <input checked="" type="checkbox"/> | Connections in idle state                          | <input checked="" type="checkbox"/> | Connections in idle state            | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30   |
| <input checked="" type="checkbox"/> | Database size in server                            | <input checked="" type="checkbox"/> | Database size in server              | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30   |
| <input checked="" type="checkbox"/> | Highest table bloat in server                      | <input checked="" type="checkbox"/> | Highest table bloat in server        | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30   |
| <input checked="" type="checkbox"/> | Largest index by table-size percentage             | <input checked="" type="checkbox"/> | Largest index by table-size perc...  | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30   |
| <input checked="" type="checkbox"/> | Last AutoVacuum                                    | <input checked="" type="checkbox"/> | Last AutoVacuum                      | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30   |
| <input checked="" type="checkbox"/> | Last Vacuum  | <input checked="" type="checkbox"/> | Last Vacuum                          | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30   |
| <input checked="" type="checkbox"/> | Log config mismatch                                | <input checked="" type="checkbox"/> | Log config mismatch                  | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30   |
| <input checked="" type="checkbox"/> | Number of prepared transactions                    | <input checked="" type="checkbox"/> | Number of prepared transactions      | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30   |
| <input checked="" type="checkbox"/> | Number of WAL archives pending                     | <input checked="" type="checkbox"/> | Number of WAL archives pending       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30   |
| <input checked="" type="checkbox"/> | Number of WAL files                                | <input checked="" type="checkbox"/> | Number of WAL files                  | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30   |
| <input checked="" type="checkbox"/> | Server Down  | <input checked="" type="checkbox"/> | Server Down                          | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30   |
| <input checked="" type="checkbox"/> | Table_Size_Alerts                                  | <input type="checkbox"/>            | Table size in server                 | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30   |
| <input checked="" type="checkbox"/> | Table size in server                               | <input checked="" type="checkbox"/> | Table size in server                 | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30   |
| <input checked="" type="checkbox"/> | Total connections                                  | <input checked="" type="checkbox"/> | Total connections                    | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30   |
| <input checked="" type="checkbox"/> | Total connections as percentage of max_con...      | <input checked="" type="checkbox"/> | Total connections as percentag...    | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30   |
| <input checked="" type="checkbox"/> | Total table bloat in server                        | <input checked="" type="checkbox"/> | Total table bloat in server          | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30   |

You can modify some properties of an existing alert in the **Alerts** table:

- The **Name** column displays the name of the alert; to change the alert name, simply replace the name in the table, and click the save icon.
- The **Auto created?** column indicates if the alert definition was automatically created; **Yes** indicates that the alert was created by PEM, and **No** indicates that the alert was manually created.
- The **Template** column displays the name of the alert template that specifies properties used by the alert. You can use the drop-down listbox to change the alert template associated with an alert.
- Use the **Enable?** switch to specify if an alert is enabled (Yes) or disabled (No).
- Use the **Interval** column to specify how often PEM should check to see if the alert conditions are satisfied. Set the **Default** switch to **No** and specify an alternate value (in **Minutes**), or return the **Default** switch to **Yes** to reset the value to its default setting. By default, PEM will check the status of each alert once every minute.
- Use the **History retention** field to specify the number of days that PEM will store data collected by the alert. Set the **Default** switch to **No** and specify an alternate value (in **Days**), or return the **Default** switch to **Yes** to reset the value to its default setting. By default, PEM will recommend storing historical data for 30 days.

Click the **Edit** icon to the left of an alert name to open the **Alert details** editor and access the complete alert definition. After modifying an alert in the editor, click the **Save** button to make your changes persistent.

## Deleting an Alert

To mark an alert for deletion, highlight the alert name in the **Alerts** table and click the delete icon to the left of the name; the alert will remain in the list, but in red strike-through font.

| Alerts                              |  |                                     |                                      |                                     |                                     |         |                                     |               |
|-------------------------------------|--|-------------------------------------|--------------------------------------|-------------------------------------|-------------------------------------|---------|-------------------------------------|---------------|
|                                     |  |                                     |                                      |                                     |                                     |         |                                     |               |
|                                     | Name   | Auto created?                       | Template                             | Enable?                             | Interval                            |         | History retention                   |               |
|                                     |  |                                     |                                      |                                     | Default?                            | Minutes | Default?                            | Days          |
| <input checked="" type="checkbox"/> | Audit config mismatch                              | <input checked="" type="checkbox"/> | Audit config mismatch                | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30            |
| <input checked="" type="checkbox"/> | Average table bloat in server                      | <input checked="" type="checkbox"/> | Average table bloat in server        | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30            |
| <input checked="" type="checkbox"/> | Connections in idle-in-transaction state           | <input checked="" type="checkbox"/> | Connections in idle-in-transactio... | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30            |
| <input checked="" type="checkbox"/> | Connections in idle-in-transaction state, as a ... | <input checked="" type="checkbox"/> | Connections in idle-in-transactio... | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30            |
| <input checked="" type="checkbox"/> | Connections in idle state                          | <input checked="" type="checkbox"/> | Connections in idle state            | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30            |
| <input checked="" type="checkbox"/> | Database size in server                            | <input checked="" type="checkbox"/> | Database size in server              | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30            |
| <input checked="" type="checkbox"/> | Highest table bloat in server                      | <input checked="" type="checkbox"/> | Highest table bloat in server        | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30            |
| <input checked="" type="checkbox"/> | Largest index by table-size percentage             | <input checked="" type="checkbox"/> | Largest index by table-size perc...  | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30            |
| <input checked="" type="checkbox"/> | Last AutoVacuum                                    | <input checked="" type="checkbox"/> | Last AutoVacuum                      | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30            |
| <input checked="" type="checkbox"/> | Last Vacuum  | <input checked="" type="checkbox"/> | Last Vacuum                          | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30            |
| <input checked="" type="checkbox"/> | Log config mismatch                                | <input checked="" type="checkbox"/> | Log config mismatch                  | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30            |
| <input checked="" type="checkbox"/> | Number of prepared transactions                    | <input checked="" type="checkbox"/> | Number of prepared transactions      | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30            |
| <input checked="" type="checkbox"/> | Number of WAL archives pending                     | <input checked="" type="checkbox"/> | Number of WAL archives pending       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30            |
| <input checked="" type="checkbox"/> | Number of WAL files                                | <input checked="" type="checkbox"/> | Number of WAL files                  | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30            |
| <input checked="" type="checkbox"/> | Server Down  | <input checked="" type="checkbox"/> | Server Down                          | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30            |
| <input checked="" type="checkbox"/> | <del>Table Size Alerts</del>                       | <input type="checkbox"/>            | <del>Table size in server</del>      | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | <del>30</del> |
| <input checked="" type="checkbox"/> | Table size in server                               | <input checked="" type="checkbox"/> | Table size in server                 | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30            |
| <input checked="" type="checkbox"/> | Total connections                                  | <input checked="" type="checkbox"/> | Total connections                    | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30            |
| <input checked="" type="checkbox"/> | Total connections as percentage of max_con...      | <input checked="" type="checkbox"/> | Total connections as percentag...    | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30            |
| <input checked="" type="checkbox"/> | Total table bloat in server                        | <input checked="" type="checkbox"/> | Total table bloat in server          | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1       | <input checked="" type="checkbox"/> | 30            |

The delete icon acts as a toggle; you can undo the deletion by clicking the delete icon a second time; when you click the Save icon, the alert definition will be permanently deleted.

## Example

The screen shown below defines an alert (named System Usage High) that monitors the committed transactions on the system:



General

Notification

Script execution

Name

System Usage High

Description

The average space wasted by tables on host, in MB.  
Required probe(s): table\_bloat,settings

Template

Average table bloat on host

A template uses metrics to generate a value to which PEM compares user specified alert boundaries. If the value returned by the template function evaluates to a value that is within the boundary of a user defined alert, PEM raises an alert.

Enable?

Yes

Select Yes to enable the alert, and No to disable the alert.

Interval

Default?

Yes

Minutes

1

Use fields in the Interval box to specify how often the alert should confirm that alert conditions are satisfied.

History retention

Default?

Yes

Days

30

Use fields in the History retention box to specify the number of days that PEM will store data collected by the alert.

Threshold values

Operator

<

Low

20

Medium

10

High

5

Unit

MB

To re-create this example, highlight the name of a PEM Agent in the tree-control, and select **Manage Alerts...** from the PEM client **Management** menu. When the **Manage Alerts** tab opens, click the add icon (+) in the upper-right hand corner of the **Alerts** table to open the alert editor.

Fields on the **General** tab instruct PEM to use the Disk busy percentage template to create the alert. The PEM server will check the free memory available once every minute, and:

- Trigger a low-severity alert if the free memory available drops below 20%
- Trigger a medium-severity alert if the free memory available drops below 10%
- Trigger a high-severity alert if the free memory available drops below 5%

General

Notification

Script execution

Email

Webhook

SNMP

Nagios

To configure notification for an alert use the below fields to specify one or multiple user group that will receive an email notification if the alert is triggered at the specific level. Use the dropdown list box to select pre-defined group that will be sent an alert of the selected level is triggered. Please note that you must configure the PEM Server to use an SMTP server to deliver email before PEM can send email notifications.

Alert types

All alerts?

No

<Default>

Low alerts?

No

<Default>

Medium alerts?

No

<Default>

High alerts?

No

<Default>

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

499

General Notification **Script execution**

**Script execution**

Execute script? ☒ Yes ☐ No Execute on alert cleared? ☐ Yes ☒ No

Execute script on ☐ PEM Server ☒ Monitored Server

Code

```
#!/bin/sh

echo "%AlertID% - %AlertName%"

export agentID="%AlertID%"

sh /tmp/notify_third_party.sh "%CurrentValue%" "%AlertRaisedTime%"
```

Use the fields in the Script execution box to (optionally) define a script that will be executed if an alert is triggered, and to specify details about the script execution.

- Set the Execute script slider to Yes to instruct PEM to execute the provided script if an alert is triggered.
- Set the Execute on alert cleared slider to Yes to instruct PEM to execute the provided script when the situation that triggered the alert has been resolved.
- Use the selector to indicate if the script should execute on the PEM Server or the Monitored Server.
- Provide the script that PEM should execute in the Code field. You can provide a batch/shell script, or SQL code. Within the script you can use the placeholders to replace the following;
  - %AlertID% - the id of the triggered alert.
  - %AlertName% - the name of the triggered alert.
  - %ObjectName% - the name of the server or agent on which the alert was triggered.
  - %ObjectType% - the type on which alert was generated.
  - %ThresholdValue% - the threshold value reached by the metric when the alert triggered.
  - %CurrentValue% - the current value of the metric that triggered the alert.
  - %CurrentState% - the current state of the alert.
  - %OldState% - the previous state of the alert.
  - %AlertRaisedTime% - the time that the alert was raised, or the most recent time that the alert state was changed.
  - %AgentID% - the id of the agent by which alert was generated.
  - %AgentName% - the name of the agent by which alert was generated.
  - %ServerID% - the id of the server on which alert was generated.
  - %ServerName% - the name of the server on which alert was generated.
  - %ServerIP% - the ip or address of the server on which alert was generated.
  - %ServerPort% - the the port of the server on which alert was generated.
  - %DatabaseName% - the name of the database on which alert was generated.
  - %SchemaName% - the name of the schema on which alert was generated.
  - %PackageName% - the name of the package on which alert was generated.
  - %DatabaseObjectName% - the name of the database object name like table name, function name etc on which alert was generated.
  - %Parameters% - the list of custom parameters used to generate the alert.
  - %AlertInfo% - the detailed database object level information of the alert.

Fields on the **Notifications** tab instruct PEM to:

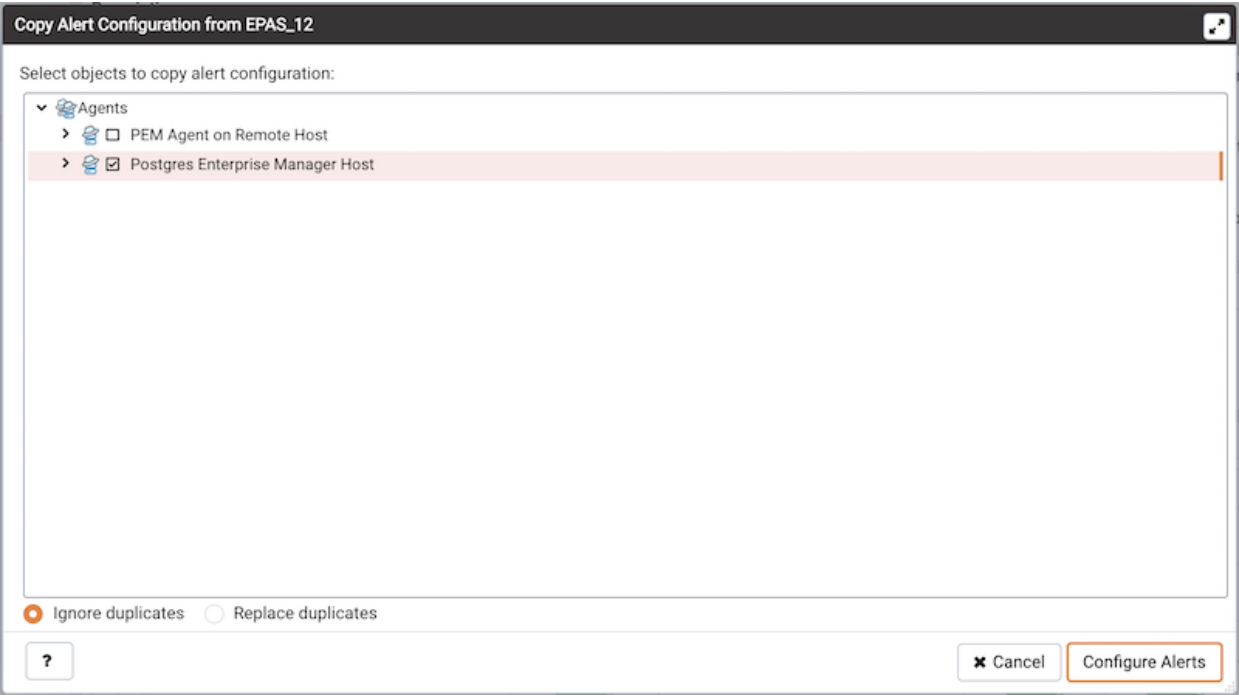
- Send an email notification to the **administrator** email group.
- Submit a passive service check result to Nagios.
- Execute the script shown in the **Code** field when the alert is triggered.
- To invoke a script on a Linux system, you must modify the entry for **batch\_script\_user** parameter of agent.cfg file and specify the user that should be used to run the script. You can either specify a non-root user or root for this parameter. If you do not specify a user, or the specified user does not exist, then the script will not be executed. Restart the agent after modifying the file. If pemagent is being run by a non-root user then the value of **batch\_script\_user** will be ignored and the script will be executed by the same non-root user that is being used for running the pemagent.
- To invoke a script on a Windows system, set the registry entry for **AllowBatchJobSteps** to true and restart the PEM agent. PEM registry entries are located in **HKEY\_LOCAL\_MACHINE\Software\EnterpriseDB\PEM\agent**.

Click the edit icon to close the editor and add the example to the **Alert List**; click the save icon before closing the **Manage Alerts** tab to save your work.

### 34.4.9.2 Copy Alerts

To speed up the deployment of alerts in the PEM system, you can copy alert definitions from one object to one or more target objects.

To copy alerts from an object, highlight the object name (from which you will copy alerts) in the PEM client tree control, and select the **Manage Alerts...** option from the **Management** menu. When the **Manage Alerts** tab opens, click the **Copy Alerts** icon (located on the **Quick Links** toolbar) to open the **Copy Alert Configuration** dialog.



The **Copy Alert Configuration** dialog copies all alerts from the object highlighted in the PEM client tree control to the object or objects selected on the dialog. Expand the tree control to select a node or nodes to specify the target object(s). Please note that the tree control displays a red warning indicator next to the source object.

To copy alerts to multiple objects at once, select a parent node of the targets. For example, to copy the alerts from one table to all tables in a schema, you can simply select the checkbox next to the schema name. PEM will only copy alerts to targets that are of the same type as the source object.

Check the **Ignore duplicates** radio button to prevent PEM from updating any existing alerts on the target objects with the same name as those being copied. Check the **Replace duplicates** radio button to replace existing alerts with alerts of the same name from the source object.

Click the **Configure Alerts** button to proceed to copy the alerts from the source object to all objects of the same type in, or below those objects selected on the **Copy Alert Configuration** dialog. When the copy is complete, a popup will notify you that the alerts have been copied to the selected target(s).

### 34.4.9.3 Alert Templates

An alert definition contains a system-defined or user-defined set of conditions that PEM compares to the system statistics; if the statistics deviate from the boundaries specified for that statistic, the alert triggers, and the PEM client displays a warning on the **Alerts Overview** page, and optionally sends a notification to a monitoring user.

The table below lists the system-defined alert templates that you can use to create an alert; please note that this list is subject to change, and may vary by system.

Within the table, the alerts are sorted by the target of the alert. The **Template Name** and **Description** columns identify and describe the behavior of the template. If the template **Details** column specifies **Yes**, metrics returned by the alert and alert parameters (if applicable) are accessible in the **Alerts table** on the **Global Overview** or **Alerts** dashboards. If the **Autocreated** column specifies **Yes**, it is the predefined alert and enabled by default:

Templates applicable on Agent

| Template Name  | Description   | Details | Autocreated |
|--|---|---------|-------------|
| Load Average (1 minute)  | 1-minute system load average.   |         |             |
| Load Average (5 minutes)   | 5-minute system load average.   |         |             |
| Load Average (15 minutes)  | 15-minute system load average.  |         |             |
| Load Average per CPU Core (1 minutes)                                  | 1-minute system load average per CPU core.                                  |         |             |
| Load Average per CPU Core (5 minutes)                                  | 5-minute system load average per CPU core.                                  |         | Yes         |
| Load Average per CPU Core (15 minutes)                                 | 15-minute system load average per CPU core.                                 |         |             |
| CPU utilization  | Average CPU consumption.  |         | Yes         |
| Number of CPUs running higher than a threshold                         | Number of CPUs running at greater than K% utilization.                      | Yes     |             |
| Free memory percentage   | Free memory as a percent of total system memory.                            |         |             |
| Memory used percentage   | Percentage of memory used.  |         | Yes         |
| Swap consumption   | Swap space consumed (in megabytes).   |         |             |
| Swap consumption percentage  | Percentage of swap area consumed.   |         | Yes         |
| Disk Consumption   | Disk space consumed (in megabytes).   |         |             |
| Disk consumption percentage  | Percentage of disk consumed.  |         |             |
| Disk Available   | Disk space available (in megabytes).  |         |             |
| Disk busy percentage   | Percentage of disk busy.  |         |             |
| Most used disk percentage  | Percentage used of the most utilized disk on the system.                    | Yes     | Yes         |
| Total table bloat on host  | The total space wasted by tables on a host, in MB.                          |         |             |
| Highest table bloat on host  | The most space wasted by a table on a host, in MB.                          |         |             |
| Average table bloat on host  | The average space wasted by tables on host, in MB.                          |         |             |
| Table size on host   | The size of tables on host, in MB.  |         |             |
| Database size on host  | The size of databases on host, in MB.                                       |         |             |
| Number of ERRORS in the logfile on agent N in last X hours             | The number of ERRORS in the logfile on agent N in last X hours.             |         |             |
| Number of WARNINGS in the logfile on agent N in last X hours           | The number of WARNINGS in the logfile on agent N in last X hours.           |         |             |
| Number of WARNINGS or ERRORS in the logfile on agent N in last X hours | The number of WARNINGS or ERRORS in the logfile on agent N in last X hours. |         |             |
| Package version mismatch   | Check for package version mismatch as per catalog.                          | Yes     |             |
| Total materialized view bloat on host                                  | The total space wasted by materialized views on a host, in MB.              |         |             |
| Highest materialized view bloat on host                                | The most space wasted by a materialized view on a host, in MB.              |         |             |
| Average materialized view bloat on host                                | The average space wasted by materialized views on host, in MB.              |         |             |
| Materialized view size on host   | The size of materialized views on host, in MB.                              |         |             |
| Agent Down   | Specified agent is currently down.  |         | Yes         |

Templates applicable on Server

| Template Name                                 | Description  | Details | Autocreated |
|---|--|---------|-------------|
| Total table bloat in server                   | The total space wasted by tables in server, in MB.   |         |             |
| Largest table (by multiple of unbloated size) | Largest table in server, calculated as a multiple of its own estimated unbloated size; exclude tables smaller than N MB. |         |             |

| Template Name   | Description   | De<br>tail<br>s | Auto<br>creat<br>ed |
|---|---|-----------------|---------------------|
| Highest table bloat in server   | The most space wasted by a table in server, in MB.  | Yes             |                     |
| Average table bloat in server   | The average space wasted by tables in server, in MB.  |                 |                     |
| Table size in server  | The size of tables in server, in MB.  | Yes             |                     |
| Database size in server   | The size of databases in server, in MB.   | Yes             |                     |
| Number of WAL files   | Total number of Write Ahead Log files.  |                 | Yes                 |
| Number of prepared transactions   | Number of transactions in prepared state.   |                 | Yes                 |
| Total connections   | Total number of connections in the server.  | Yes             | Yes                 |
| Total connections as percentage of max_connections                        | Total number of connections in the server as a percentage of maximum connections allowed on server, settings.                                   |                 | Yes                 |
| Unused, non-superuser connections   | Number of unused, non-superuser connections on the server, user_info, settings.   |                 |                     |
| Unused, non-superuser connections as percentage of max_connections        | Number of unused, non-superuser connections on the server as a percentage of max_connections, user_info, settings.                              |                 |                     |
| Ungranted locks   | Number of ungranted locks in server.  |                 |                     |
| Percentage of buffers written by backends                                 | The percentage of buffers written by backends vs. the total buffers written.  |                 |                     |
| Percentage of buffers written by checkpoint                               | The percentage of buffers written by the checkpoints vs. the total buffers written.   |                 |                     |
| Buffers written per second  | Number of buffers written per second, over the last two probe cycles.   |                 |                     |
| Buffers allocated per second  | Number of buffers allocated per second, over the last two probe cycles.   |                 |                     |
| Connections in idle state   | Number of connections in server that are in idle state.   | Yes             | Yes                 |
| Connections in idle-in-transaction state                                  | Number of connections in server that are in idle-in-transaction state.  | Yes             | Yes                 |
| Connections in idle-in-transaction state,as percentage of max_connections | Number of connections in server that are in idle-in-transaction state, as a percentage of maximum connections allowed on server, settings       |                 | Yes                 |
| Long-running idle connections   | Number of connections in the server that have been idle for more than N seconds.  | Yes             |                     |
| Long-running idle connections and idle transactions                       | Number of connections in the server that have been idle or idle-in-transaction for more than N seconds.   | Yes             |                     |
| Long-running idle transactions  | Number of connections in the server that have been idle in transaction for more than N seconds.   | Yes             |                     |
| Long-running transactions   | Number of transactions in server that have been running for more than N seconds.  | Yes             |                     |
| Long-running queries  | Number of queries in server that have been running for more than N seconds. It does not include the long running vacuum or auto vacuum queries. | Yes             |                     |
| Long-running vacuums  | Number of vacuum operations in server that have been running for more than N seconds.   | Yes             |                     |
| Long-running autovacuids  | Number of autovacuum operations in server that have been running for more than N seconds.   | Yes             |                     |
| Committed transactions percentage   | Percentage of transactions in the server that committed vs. that rolled-back over last N minutes.   |                 |                     |
| Shared buffers hit percentage   | Percentage of block read requests in the server that were satisfied by shared buffers, over last N minutes.                                     |                 |                     |
| Tuples inserted   | Tuples inserted into server over last N minutes.  |                 |                     |
| InfiniteCache buffers hit percentage                                      | Percentage of block read requests in the server that were satisfied by InfiniteCache, over last N minutes.                                      |                 |                     |
| Tuples fetched  | Tuples fetched from server over last N minutes.   |                 |                     |
| Tuples returned   | Tuples returned from server over last N minutes.  |                 |                     |
| Dead Tuples   | Number of estimated dead tuples in server.  |                 |                     |
| Tuples updated  | Tuples updated in server over last N minutes.   |                 |                     |
| Tuples deleted  | Tuples deleted from server over last N minutes.   |                 |                     |
| Tuples hot updated  | Tuples hot updated in server, over last N minutes.  |                 |                     |
| Sequential Scans  | Number of full table scans in server, over last N minutes.  |                 |                     |

| Template Name   | Description  | De<br>tail<br>s | Auto<br>creat<br>ed |
|---|--|-----------------|---------------------|
| Index Scans   | Number of index scans in server, over last N minutes.  |                 |                     |
| Hot update percentage   | Percentage of hot updates in the server over last N minutes.   |                 |                     |
| Live Tuples   | Number of estimated live tuples in server.   |                 |                     |
| Dead tuples percentage  | Percentage of estimated dead tuples in server.   |                 |                     |
| Last Vacuum   | Hours since last vacuum on the server.   |                 | Yes                 |
| Last AutoVacuum   | Hours since last autovacuum on the server.   |                 | Yes                 |
| Last Analyze  | Hours since last analyze on the server.  |                 |                     |
| Last AutoAnalyze  | Hours since last autoanalyze on the server.  |                 |                     |
| Percentage of buffers written by backends over last N minutes               | The percentage of buffers written by backends vs. the total buffers written over last N minutes.   |                 |                     |
| Table Count   | Total number of tables in server.  |                 |                     |
| Function Count  | Total number of functions in server.   |                 |                     |
| Sequence Count  | Total number of sequences in server.   |                 |                     |
| A user expires in N days  | Number of days before a user's validity expires.   |                 |                     |
| Index size as a percentage of table size                                    | Size of the indexes in server, as a percentage of their tables' size.  |                 |                     |
| Largest index by table-size percentage                                      | Largest index in server, calculated as percentage of its table's size, oc_index, table_size.   |                 | Yes                 |
| Number of ERRORS in the logfile on server M in the last X hours             | The number of ERRORS in the logfile on server M in last X hours.   |                 |                     |
| Number of WARNINGS in the logfile on server M in the last X hours           | The number of WARNINGS in logfile on server M in the last X hours.   |                 |                     |
| Number of WARNINGS or ERRORS in the logfile on server M in the last X hours | The number of WARNINGS or ERRORS in the logfile on server M in the last X hours.   |                 |                     |
| Number of attacks detected in the last N minutes                            | The number of SQL injection attacks occurred in the last N minutes.  |                 |                     |
| Number of attacks detected in the last N minutes by username                | The number of SQL injection attacks occurred in the last N minutes by username.  |                 |                     |
| Number of replica servers lag behind the primary by write location          | Streaming Replication: number of replica servers lag behind the primary by write location.   | Yes             |                     |
| Number of replica servers lag behind the primary by flush location          | Streaming Replication: number of replica servers lag behind the primary by flush location.   | Yes             |                     |
| Number of replica servers lag behind the primary by replay location         | Streaming Replication: number of replica servers lag behind the primary by replay location.  | Yes             |                     |
| Replica server lag behind the primary by write location                     | Streaming Replication: replica server lag behind the primary by write location in MB.  | Yes             |                     |
| Replica server lag behind the primary by flush location                     | Streaming Replication: replica server lag behind the primary by flush location in MB.  | Yes             |                     |
| Replica server lag behind the primary by WAL pages                          | Streaming Replication: replica server lag behind the primary by WAL pages.   |                 |                     |
| Replica server lag behind the primary by WAL segments                       | Streaming Replication: replica server lag behind the primary by WAL segments.  |                 |                     |
| Replica server lag behind the primary by replay location                    | Streaming Replication: replica server lag behind the primary by replay location in MB.   | Yes             |                     |
| Replica server lag behind the primary by size (MB)                          | Streaming Replication: replica server lag behind the primary by size in MB.  | Yes             |                     |
| Total materialized view bloat in server                                     | The total space wasted by materialized views in server, in MB.   |                 |                     |
| Largest materialized view (by multiple of unbloated size)                   | Largest materialized view in server, calculated as a multiple of its own estimated unbloated size; exclude materialized views smaller than N MB. |                 |                     |
| Highest materialized view bloat in server                                   | The most space wasted by a materialized view in server, in MB.   |                 |                     |

| Template Name   | Description   | De<br>tail<br>s | Auto<br>creat<br>ed |
|---|---|-----------------|---------------------|
| Average materialized view bloat in server                   | The average space wasted by materialized views in server, in MB.                          |                 |                     |
| Materialized view size in server                            | The size of materialized view in server, in MB.   |                 |                     |
| View Count  | Total number of views in server.  |                 |                     |
| Materialized View Count                                     | Total number of materialized views in server.   |                 |                     |
| Audit config mismatch                                       | Check for audit config parameter mismatch   | Yes             | Yes                 |
| Server Down   | Specified server is currently inaccessible.   |                 | Yes                 |
| Number of WAL archives pending                              | Streaming Replication: number of WAL files pending to be replayed at replica.             |                 |                     |
| Number of minutes lag of replica server from primary server | Streaming Replication: number of minutes replica node is lagging behind the primary node. |                 |                     |
| Log config mismatch   | Check for log config parameter mismatch.  | Yes             | Yes                 |
| PGD Group Raft Consensus                                    | PGD group raft consensus not working  | Yes             | Yes                 |
| PGD Group Raft Leader ID not matching                       | PGD group raft leader id not matching   | Yes             | Yes                 |
| PGD Group Versions check                                    | PGD/pglogical version mismatched in PGD group   | Yes             | Yes                 |
| PGD worker error detected                                   | PGD worker error detected reported for PGD node   |                 | Yes                 |
| Transaction ID exhaustion (wraparound)                      | Check for Transaction ID exhaustion (wraparound).   | Yes             | Yes                 |

Templates applicable on Database

| Template Name   | Description  | De<br>tail<br>s | Auto<br>creat<br>ed |
|---|--|-----------------|---------------------|
| Total table bloat in database   | The total space wasted by tables in database, in MB.   |                 |                     |
| Largest table (by multiple of unbloated size)                             | Largest table in database, calculated as a multiple of its own estimated unbloated size; exclude tables smaller than N MB.                   |                 |                     |
| Highest table bloat in database   | The most space wasted by a table in database, in MB.   |                 |                     |
| Average table bloat in database   | The average space wasted by tables in database, in MB.   |                 |                     |
| Table size in database  | The size of tables in database, in MB.   | Yes             |                     |
| Database size   | The size of the database, in MB.   |                 |                     |
| Total connections   | Total number of connections in the database.   | Yes             |                     |
| Total connections as percentage of max_connections                        | Total number of connections in the database as a percentage of maximum connections allowed on server, settings.                              |                 |                     |
| Ungranted locks   | Number of ungranted locks in database.   |                 |                     |
| Connections in idle state   | Number of connections in database that are in idle state.  | Yes             |                     |
| Connections in idle-in-transaction state                                  | Number of connections in database that are in idle-in-transaction state.   | Yes             |                     |
| Connections in idle-in-transaction state,as percentage of max_connections | Number of connections in database that are in idle-in-transaction state, as a percentage of maximum connections allowed on server, settings. |                 |                     |
| Long-running idle connections   | Number of connections in the database that have been idle for more than N seconds.   | Yes             |                     |
| Long-running idle connections and idle transactions                       | Number of connections in the database that have been idle or idle-in-transaction for more than N seconds.                                    | Yes             |                     |
| Long-running idle transactions  | Number of connections in the database that have been idle in transaction for more than N seconds.  | Yes             |                     |
| Long-running transactions   | Number of transactions in database that have been running for more than N seconds.   | Yes             |                     |

| Template Name  | Description   | De<br>tail<br>s | Auto<br>creat<br>ed |
|--|---|-----------------|---------------------|
| Long-running queries   | Number of queries in database that have been running for more than N seconds. It does not include the long running vacuum or auto vacuum queries. | Yes             |                     |
| Long-running vacuums   | Number of vacuum operations in database that have been running for more than N seconds.   | Yes             |                     |
| Long-running autovacuum                                      | Number of autovacuum operations in database that have been running for more than N seconds.   | Yes             |                     |
| Committed transactions percentage                            | Percentage of transactions in the database that committed vs. that rolled-back over last N minutes.   |                 |                     |
| Shared buffers hit percentage                                | Percentage of block read requests in the database that were satisfied by shared buffers, over last N minutes.                                     |                 |                     |
| InfiniteCache buffers hit percentage                         | Percentage of block read requests in the database that were satisfied by InfiniteCache, over last N minutes.                                      |                 |                     |
| Tuples fetched   | Tuples fetched from database over last N minutes.   |                 |                     |
| Tuples returned  | Tuples returned from database over last N minutes.  |                 |                     |
| Tuples inserted  | Tuples inserted into database over last N minutes.  |                 |                     |
| Tuples updated   | Tuples updated in database over last N minutes.   |                 |                     |
| Tuples deleted   | Tuples deleted from database over last N minutes.   |                 |                     |
| Tuples hot updated   | Tuples hot updated in database, over last N minutes.  |                 |                     |
| Sequential Scans   | Number of full table scans in database, over last N minutes.  |                 |                     |
| Index Scans  | Number of index scans in database, over last N minutes.   |                 |                     |
| Hot update percentage  | Percentage of hot updates in the database over last N minutes.  |                 |                     |
| Live Tuples  | Number of estimated live tuples in database.  |                 |                     |
| Dead Tuples  | Number of estimated dead tuples in database.  |                 |                     |
| Dead tuples percentage                                       | Percentage of estimated dead tuples in database.  |                 |                     |
| Last Vacuum  | Hours since last vacuum on the database.  |                 |                     |
| Last AutoVacuum  | Hours since last autovacuum on the database.  |                 |                     |
| Last Analyze   | Hours since last analyze on the database.   |                 |                     |
| Last AutoAnalyze   | Hours since last autoanalyze on the database.   |                 |                     |
| Table Count  | Total number of tables in database.   |                 |                     |
| Function Count   | Total number of functions in database.  |                 |                     |
| Sequence Count   | Total number of sequences in database.  |                 |                     |
| Index size as a percentage of table size                     | Size of the indexes in database, as a percentage of their tables' size.   |                 |                     |
| Largest index by table-size percentage                       | Largest index in database, calculated as percentage of its table's size, oc_index, table_size   |                 |                     |
| Database Frozen XID  | The age (in transactions before the current transaction) of the database's frozen transaction ID.   |                 |                     |
| Number of attacks detected in the last N minutes             | The number of SQL injection attacks occurred in the last N minutes.   |                 |                     |
| Number of attacks detected in the last N minutes by username | The number of SQL injection attacks occurred in the last N minutes by username.   |                 |                     |
| Queries that have been cancelled due to dropped tablespaces  | Streaming Replication: number of queries that have been cancelled due to dropped tablespaces.   |                 |                     |
| Queries that have been cancelled due to lock timeouts        | Streaming Replication: number of queries that have been cancelled due to lock timeouts.   |                 |                     |
| Queries that have been cancelled due to old snapshots        | Streaming Replication: number of queries that have been cancelled due to old snapshots.   |                 |                     |
| Queries that have been cancelled due to pinned buffers       | Streaming Replication: number of queries that have been cancelled due to pinned buffers.  |                 |                     |



| Template Name   | Description  | De<br>tail<br>s | Auto<br>creat<br>ed |
|---|--|-----------------|---------------------|
| Queries that have been cancelled due to deadlocks         | Streaming Replication: number of queries that have been cancelled due to deadlocks.  |                 |                     |
| Total events lagging in all slony clusters                | Slony Replication: total events lagging in all slony clusters.   | Yes             |                     |
| Events lagging in one slony cluster                       | Slony Replication: events lagging in one slony cluster.  |                 |                     |
| Lag time (minutes) in one slony cluster                   | Slony Replication: lag time (minutes) in one slony cluster.  |                 |                     |
| Total rows lagging in xdb single primary replication      | xDB Replication: Total rows lagging in xdb single primary replication  | Yes             |                     |
| Total rows lagging in xdb multi primary replication       | xDB Replication: Total rows lagging in xdb multi primary replication   | Yes             |                     |
| Total materialized view bloat in database                 | The total space wasted by materialized views in database, in MB.   |                 |                     |
| Largest materialized view (by multiple of unbloated size) | Largest materialized view in database, calculated as a multiple of its own estimated unbloated size; exclude materialized views smaller than N MB. |                 |                     |
| Highest materialized view bloat in database               | The most space wasted by a materialized view in database, in MB.   |                 |                     |
| Average materialized view bloat in database               | The average space wasted by materialized views in database, in MB.   |                 |                     |
| Materialized view size in database                        | The size of materialized view in database, in MB.  |                 |                     |
| View Count  | Total number of views in database.   |                 |                     |
| Materialized View Count                                   | Total number of materialized views in database.  |                 |                     |

Templates applicable on Schema

| Template Name                                 | Description  | De<br>tail<br>s | Autocr<br>eated |
|---|--|-----------------|-----------------|
| Total table bloat in schema                   | The total space wasted by tables in schema, in MB.   | Yes             |                 |
| Largest table (by multiple of unbloated size) | Largest table in schema, calculated as a multiple of its own estimated unbloated size; exclude tables smaller than N MB. |                 |                 |
| Highest table bloat in schema                 | The most space wasted by a table in schema, in MB.   |                 |                 |
| Average table bloat in schema                 | The average space wasted by tables in schema, in MB.   |                 |                 |
| Table size in schema                          | The size of tables in schema, in MB.   | Yes             |                 |
| Tuples inserted                               | Tuples inserted in schema over last N minutes.   |                 |                 |
| Tuples updated                                | Tuples updated in schema over last N minutes.  |                 |                 |
| Tuples deleted                                | Tuples deleted from schema over last N minutes.  |                 |                 |
| Tuples hot updated                            | Tuples hot updated in schema, over last N minutes.   |                 |                 |
| Sequential Scans                              | Number of full table scans in schema, over last N minutes.   |                 |                 |
| Index Scans                                   | Number of index scans in schema, over last N minutes.  |                 |                 |
| Hot update percentage                         | Percentage of hot updates in the schema over last N minutes.   |                 |                 |
| Live Tuples                                   | Number of estimated live tuples in schema.   |                 |                 |
| Dead Tuples                                   | Number of estimated dead tuples in schema.   |                 |                 |
| Dead tuples percentage                        | Percentage of estimated dead tuples in schema.   |                 |                 |
| Last Vacuum                                   | Hours since last vacuum on the schema.   |                 |                 |
| Last AutoVacuum                               | Hours since last autovacuum on the schema.   |                 |                 |
| Last Analyze                                  | Hours since last analyze on the schema.  |                 |                 |
| Last AutoAnalyze                              | Hours since last autoanalyze on the schema.  |                 |                 |

| Template Name   | Description   | Details | Autogenerated |
|---|---|---------|---------------|
| Table Count   | Total number of tables in schema.   |         |               |
| Function Count  | Total number of functions in schema.  |         |               |
| Sequence Count  | Total number of sequences in schema.  |         |               |
| Index size as a percentage of table size                  | Size of the indexes in schema, as a percentage of their table's size.   |         |               |
| Largest index by table-size percentage                    | Largest index in schema, calculated as percentage of its table's size, oc_index, table_size   |         |               |
| Materialized View bloat                                   | Space wasted by the materialized view, in MB.   |         |               |
| Total materialized view bloat in schema                   | The total space wasted by materialized views in schema, in MB.  |         |               |
| Materialized view size as a multiple of ubloated size     | Size of the materialized view as a multiple of estimated unbloated size.  |         |               |
| Largest materialized view (by multiple of unbloated size) | Largest materialized view in schema, calculated as a multiple of its own estimated unbloated size; exclude materialized view smaller than N MB. |         |               |
| Highest materialized view bloat in schema                 | The most space wasted by a materialized view in schema, in MB.  |         |               |
| Average materialized view bloat in schema                 | The average space wasted by materialized views in schema, in MB.  |         |               |
| Materialized view size                                    | The size of materialized view, in MB.   |         |               |
| Materialized view size in schema                          | The size of materialized views in schema, in MB.  |         |               |
| View Count  | Total number of views in schema.  |         |               |
| Materialized View Count                                   | Total number of materialized views in schema.   |         |               |
| Materialized View Frozen XID                              | The age (in transactions before the current transaction) of the materialized view's frozen transaction ID.                                      |         |               |

Templates applicable on Table

| Template Name                             | Description  | Details | Autocreated |
|---|--|---------|-------------|
| Table bloat                               | Space wasted by the table, in MB.                            |         |             |
| Table size                                | The size of table, in MB.                                    |         |             |
| Table size as a multiple of ubloated size | Size of the table as a multiple of estimated unbloated size. |         |             |
| Tuples inserted                           | Tuples inserted in table over last N minutes.                |         |             |
| Tuples updated                            | Tuples updated in table over last N minutes.                 |         |             |
| Tuples deleted                            | Tuples deleted from table over last N minutes.               |         |             |
| Tuples hot updated                        | Tuples hot updated in table, over last N minutes.            |         |             |
| Sequential Scans                          | Number of full table scans on table, over last N minutes.    |         |             |
| Index Scans                               | Number of index scans on table, over last N minutes.         |         |             |
| Hot update percentage                     | Percentage of hot updates in the table over last N minutes.  |         |             |
| Live Tuples                               | Number of estimated live tuples in table.                    |         |             |
| Dead Tuples                               | Number of estimated dead tuples in table.                    |         |             |
| Dead tuples percentage                    | Percentage of estimated dead tuples in table.                |         |             |
| Last Vacuum                               | Hours since last vacuum on the table.                        |         |             |
| Last AutoVacuum                           | Hours since last autovacuum on the table.                    |         |             |

| Template Name                            | Description  | Details | Autocreated |
|--|--|---------|-------------|
| Last Analyze                             | Hours since last analyze on the table.   |         |             |
| Last AutoAnalyze                         | Hours since last autoanalyze on the table.   |         |             |
| Row Count                                | Estimated number of rows in a table.   |         |             |
| Index size as a percentage of table size | Size of the indexes on table, as a percentage of table's size.                                 |         |             |
| Table Frozen XID                         | The age (in transactions before the current transaction) of the table's frozen transaction ID. |         |             |

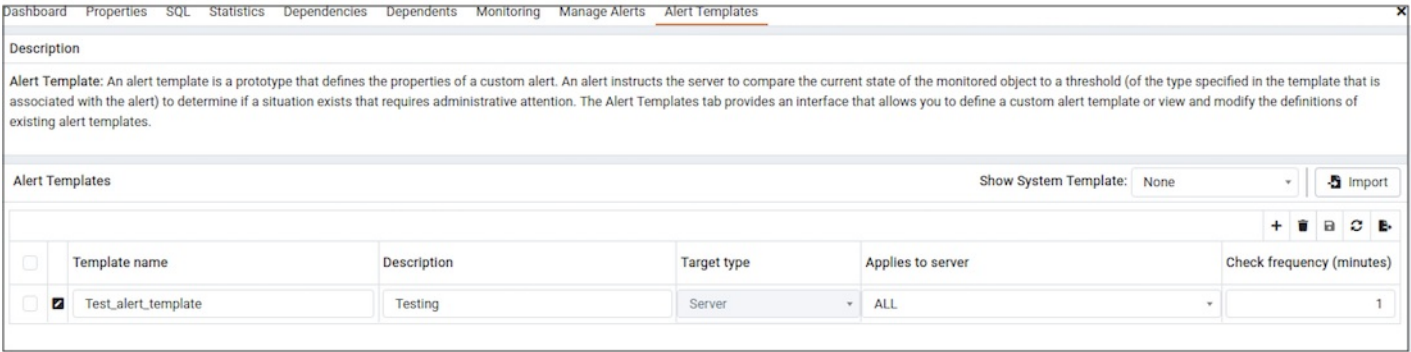
Global Templates

| Template Name | Description   | Details | Autocreated |
|---------------|---|---------|-------------|
| Agents Down   | Number of agents that haven't reported in recently. |         |             |
| Servers Down  | Number of servers that are currently inaccessible.  |         |             |
| Alert Errors  | Number of alerts in an error state.                 |         |             |

34.4.9.4 Custom Alert Templates

An alert template is a prototype that defines the properties of an [alert](#). An alert instructs the server to compare the current state of the monitored object to a threshold (specified in the alert template) to determine if a situation exists that requires administrative attention.

You can use the [Alert Templates](#) tab to define a custom alert template or view the definitions of existing alert templates. To open the [Alert Template](#) tab, select the [Manage Alerts...](#) menu option from the [Management](#) menu; when the [Manage Alerts](#) tab opens, select [Alert Templates](#) from the [Quick Links](#) menu.



Use the [Show System Template](#) drop-down listbox to filter the alert templates; select a type from the listbox to view all of the templates for that level of the PEM hierarchy.

Reviewing an Existing Alert Template

To view the definition of an existing template (including PEM pre-defined alert templates), use the [Show System Template](#) drop-down listbox to select the type of object monitored. When you select the object type, the [Alert Templates](#) table will display the currently defined alert templates that correspond with that object type. Highlight a template name and click the edit icon (at the left end of the row) to review the template definition.

Dashboard Properties SQL Statistics Dependencies Dependents Monitoring Manage Alerts Alert Templates pem/postgres...

Description

Alert Template: An alert template is a prototype that defines the properties of a custom alert. An alert instructs the server to compare the current state of the monitored object to a threshold (of the type specified in the template that is associated with the alert) to determine if a situation exists that requires administrative attention. The Alert Templates tab provides an interface that allows you to define a custom alert template or view and modify the definitions of existing alert templates.

Alert Templates Show System Template: Global Import

| <input type="checkbox"/> | Template name   | Description                                     | Target type | Applies to server | Check frequency (minutes) |
|--------------------------|---|---|-------------|-------------------|---------------------------|
| <input type="checkbox"/> | <input checked="" type="checkbox"/> Agents Down         | Number of agents that haven't reported in re... | Global      | ALL               | 1                         |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> Alert Errors        | Number of alerts in an error state.             | Global      | ALL               | 1                         |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> Servers Down        | Number of servers that are currently inacces... | Global      | ALL               | 1                         |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> Test_alert_template | Testing   | Server      | ALL               | 1                         |

Use the edit button to the left of a template name to view detailed information about the template:

- General information is displayed on the **General** tab.
- The names of probes that provide data for the template are listed on the **Probe Dependency** tab.
- The names of any parameters referred to in the SQL code are listed on the **Parameters** tab.
- The SQL code that defines the behavior of the alert is displayed on the **SQL** tab.

Defining a New Alert Template

To define a new alert template, use the **Show System Template** drop-down listbox to select **None** , and click the **Add** icon (+) located in the upper-right corner of the alert template table.

General Probe Dependency Parameters SQL

Template name

Description

Target type Server

Applies to server ALL

History retention 30

Threshold unit

Auto create

Auto create? No

Operator > Low Med High

Check frequency (minutes) 1

Use fields on the **General** tab to specify general information about the template:

- Use the **Template name** field to specify a name for the new alert template; this field is required.
- Use the **Description** field to provide a description of the alert template; this field is required.
- Use the **Target type** drop-down listbox to select the type of object that will be the focus of the alert.
- Use the **Applies to server** drop-down listbox to specify the server type (EDB Postgres Advanced Server or PostgreSQL) to which the alert will be applied; you can specify a single server type, or **ALL**.
- Use the **History retention** field to specify the number of days that the result of the alert execution will be stored on the PEM server.
- Use the **Threshold unit** field to specify a unit type of the threshold value that corresponds to the output type of the underlying SQL statement. For example, the following units are used in the system alert templates:

| Threshold unit | System Alert Template                                       |
|----------------|---|
| #              | Function count  |
| %              | CPU utilization   |
| ' '            | Events lagging in one slony cluster                         |
| days           | A user expires in N days                                    |
| Hours          | Last Vacuum   |
| Minutes        | Number of minutes lag of replica server from primary server |
| MB             | Average table bloat in server                               |
| NULL           | Swap consumption  |
| STATE          | Server down   |

#### Note

- **NULL** type is only defined for a few system alert templates and cannot be defined for the custom alert templates.
  - You can also define any meaningful **Threshold unit** such as:
    - **No of leader ID's** for **PGD Group Raft Leader ID not matching**
    - **No of worker error** for **PGD worker error detected**
- Use fields in the **Auto create** box to indicate if PEM should use the template to generate an automatic alert. If enabled, PEM will automatically create an alert when a new server or agent (as specified by the **Target type** drop-down listbox) is added, and delete that alert when the target object is dropped.
  - Move the **Auto create?** slider to **Yes** to indicate that PEM should automatically create alerts based on the template. If you modify an existing alert template, changing the **Auto create?** slider from **No** to **Yes**, PEM will create alerts on the existing agents and servers. Please note that if you change the slider from **Yes** to **No**, the default threshold values in existing alerts will be erased, and cannot be recovered.
  - Use the **Operator** drop-down listbox to select the operator that PEM will use when evaluating the current system values.
 

Select a greater-than sign (>) to indicate that the alert should be triggered when the system values are greater than the values entered in the **Threshold values** fields.

Select a less-than sign (<) to indicate that the alert should be triggered when the system values are less than the values entered in the **Threshold values** fields.
  - Use the threshold fields to specify the values that PEM will compare to the system values to determine if an alert should be raised. Please

note that you must specify values for all three thresholds ( **Low** , **Medium** , and **High** ):

- Enter a value that will trigger a low-severity alert in the **Low** field.
  - Enter a value that will trigger a medium-severity alert in the **Medium** field.
  - Enter a value that will trigger a high-severity alert in the **High** field.
- Use the **Check frequency** field to specify the default number of minutes between alert executions. This value specifies how often the server will invoke the SQL code specified in the definition and compare the result to the threshold value specified in the template.

General

Probe Dependency

Parameters

SQL

Probes

Select probe from list

+

Display name

Internal name

Use the fields on the **Probe Dependency** tab to specify the names of probes referred to in the SQL query specified on the **SQL** tab:

- Use the **Probes** drop-down listbox to select from a list of the available probes; highlight a probe name, and click the **Add** button to add the probe to the list of probes used by the alert template. To remove a probe from the selected probes list, highlight the probe name, and click the **Delete** icon.

General

Probe Dependency

Parameters

SQL

Parameters

+

Name

Data type

Unit

Add (+) button is disabled in case of system template or value of "Auto create" is Yes in General tab.

Use fields on the **Parameters** tab to define the parameters that will be used in the SQL code specified on the **SQL** tab. Click the **Add** icon, and:

- Use the **Name** field to specify the parameter name.
- Use the **Data type** drop-down listbox to select the type of parameter.
- Use the **Unit** field to specify the type of unit specified by the parameter.

When you've defined a new parameter, click the **Add/Change** button to save the definition and add the parameter to the parameter list.

To modify an existing parameter definition, highlight a parameter name in the list, modify the parameter values in the fields at the bottom of the tab, and click **Add/Change** to preserve the changes. To remove one or more parameter definitions, highlight the parameter name(s) and click the **Remove** button.

General

Probe Dependency

Parameters

SQL

Code

⚠

Use the Code field to provide the text of the SQL query that the server will invoke when executing the alert. The SQL query will provide the result against which the threshold value is compared; if the alert result deviates from the specified threshold value, an alert will be raised.

Detailed information SQL

Use the Detailed information SQL field to specify the SQL query that will provide the detailed information about that alert on Dashboard.

Use the **Code** field on the **SQL** tab to provide the text of the SQL query that the server will invoke when executing the alert. The SQL query will provide

the result against which the threshold value is compared; if the alert result deviates from the specified threshold value, an alert will be raised.

Within the query, parameters defined on the **Parameters** tab should be referenced (sequentially) by the variable **param\_x** , where **x** indicates the position of the parameter definition within the parameter list. For example, **param\_1** refers to the first parameter in the parameter list, param\_2 refers to the second parameter in the parameter list, and so on.

The query can also include the following pre-defined variables:

| Variable Description                    | Variable Name       |
|---|---------------------|
| agent identifier                        | '\${agent_id}'      |
| server identifier                       | '\${server_id}'     |
| database name                           | '\${database_name}' |
| schema name                             | '\${schema_name}'   |
| table, index, sequence or function name | '\${object_name}'   |

Please Note: If the specified query is dependent on one or more probes from different levels within the PEM hierarchy (server, database, schema, etc.), and a probe becomes disabled, any resulting alerts will be displayed as follows:

- If the alert definition and the probe referenced by the query are from the same level within the PEM hierarchy, the server will display any alerts that reference the alert template on the Alert Error table of the Global Alert Dashboard.
- If the alert definition and the probe referenced by the query are from different levels of the PEM hierarchy, the server will display any triggered alerts that reference the alert template on the Alert Details table of the hierarchy on which the Alert was defined.

Use the **Detailed Information SQL** field to provide a SQL query that will be invoked if the alert is triggered. The result set of the query may be displayed as part of the detailed alert information on the **Alerts** dashboard or **Global Overview** dashboard.

After defining a new alert template, click the **Add/Change** button to save the definition and add the template to the **Alert Templates list** . Click **Cancel** to exit the **Alert Templates** dialog without saving changes.

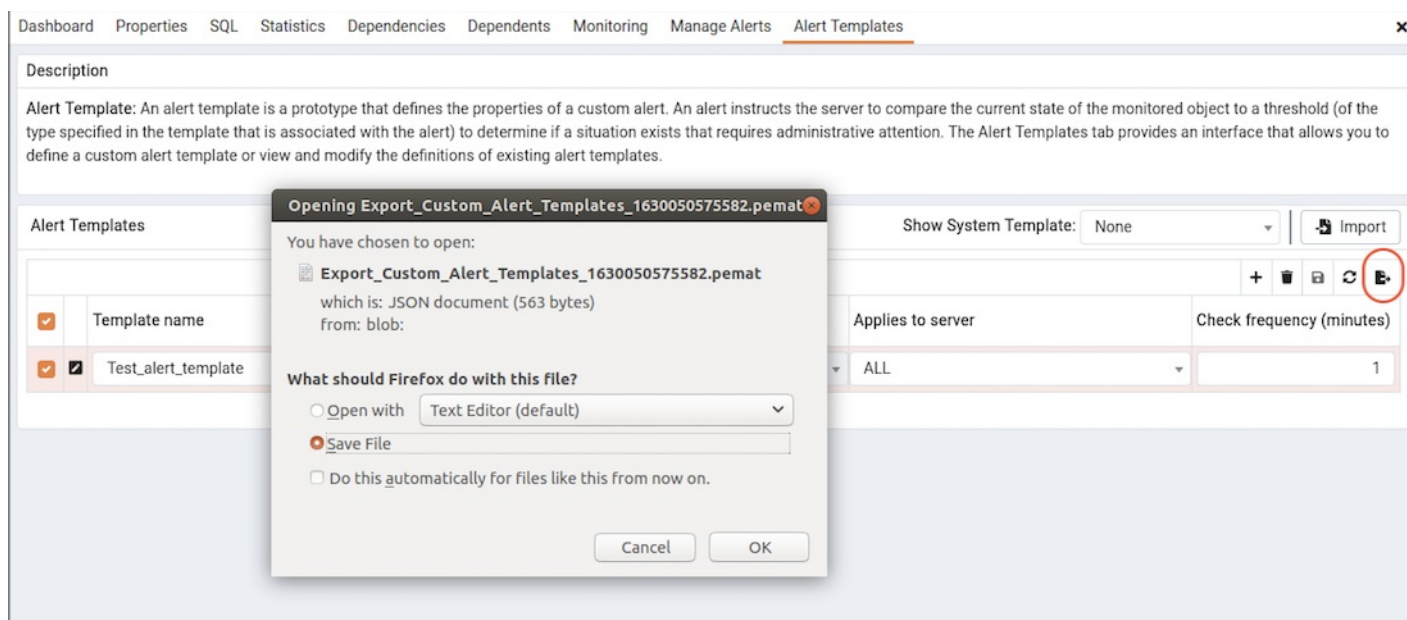
After defining a template, you can use the **Manage Alerts** tab to create and enable an alert based on the template.

Exporting or Importing an Alert Template

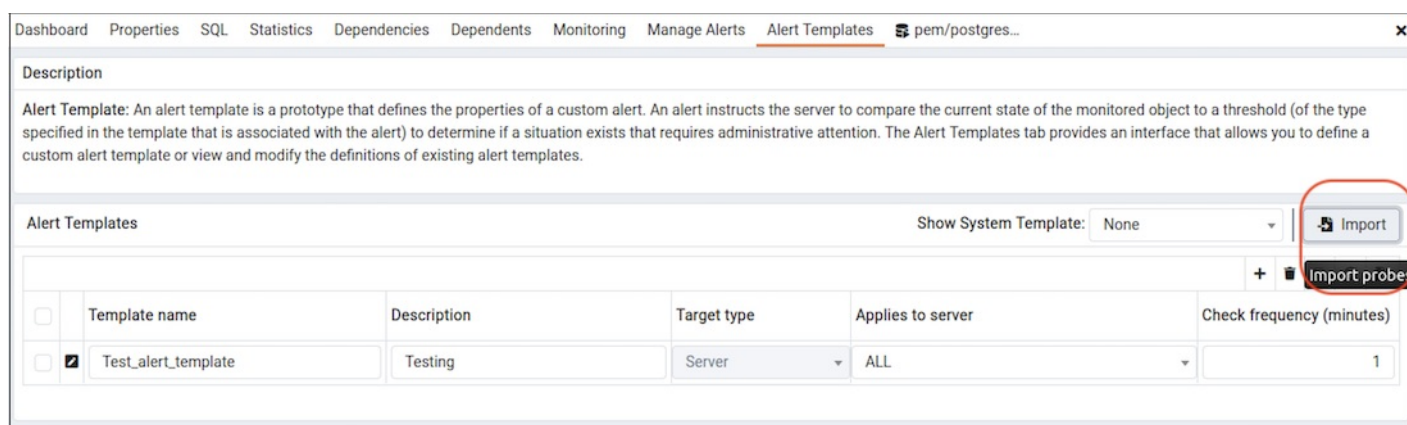
From PEM 8.2 onwards, you can export or import the alert templates to another PEM Server.

To **Export** the Alert Template, select any alert template from the **Alert Templates** tab and then select the **Export** icon in the upper-right corner of the table. Select **Save File** option and then select **ok** , it will generate the **JSON** file.

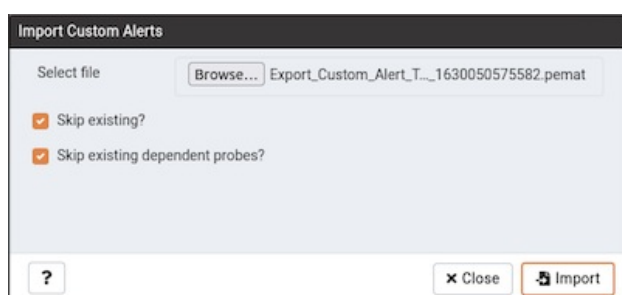




To **Import** the Alert Template, go to the **Alert Templates** tab and then select the **Import** icon in the upper-right corner.



Click on the **Browse** button to select the **JSON** file with the code to be imported and then click **Import**.

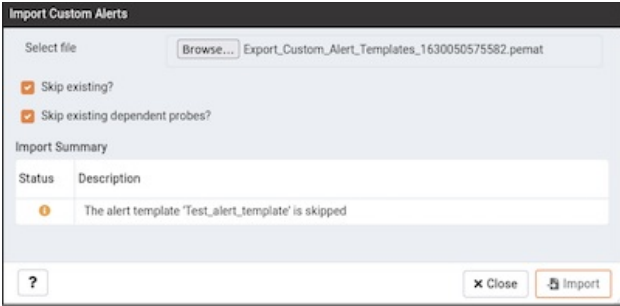


After selecting the file to import, you can select the following checkboxes:

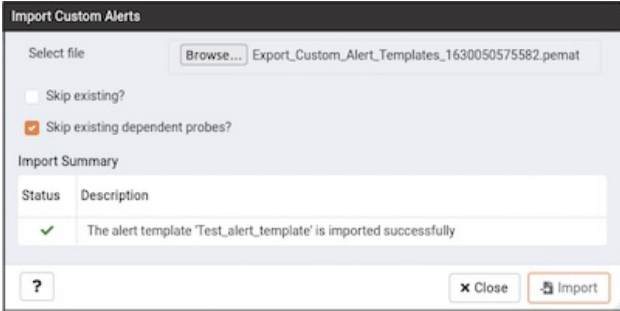
- **skip existing** - If you select this checkbox then it will skip the alert template if it already exists.
- **skip existing dependent probe** - The alert templates are dependent on probes. If you select this checkbox then it will skip the dependent probe if it already exists.

If both the checkboxes are selected and the alert template already exists, then it skips importing the alert template with below message:

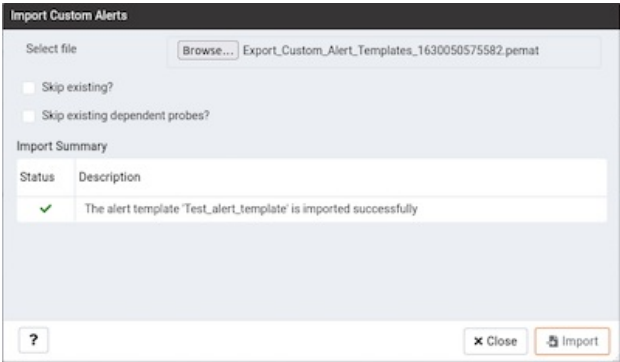




If the `skip existing` checkbox is not selected and `skip dependent probe` is selected and the alert template already exists, then it will import the alert template successfully as below:



If both the checkboxes are not selected and the alert template does not exist, then it will successfully import the alert template as below:



Deleting an Alert Template

To delete an alert template, select the template name in the alert templates table, and click the `Delete` icon (located to the upper-right corner of the table). The alert history will persist for the length of time specified on the `History Retention` field in the template definition.


34.4.9.5 Email Groups

Postgres Enterprise Manager monitors your system for conditions that require user attention. You can use an email group to specify the email addresses of users that the server will notify if current values deviate from threshold values specified in an alert definition. An email group has the flexibility to notify multiple users, or target specific users during user-defined time periods.

Please note that you must configure the PEM Server to use an SMTP server to deliver email before PEM can send email notifications.

Use the `Email Groups` tab to configure groups of SMTP email recipients. To access the `Email Groups` tab, select `Manage Alerts...` from the PEM client's `Management` menu; when the `Manage Alerts` tab opens, select `Email Groups` from the `Quick Links` toolbar.





Email Groups

|   | Group name   |
|---|--------------|
|   | acctg_admin  |
|   | hr_resources |
|   | sales        |
|   | <Default>    |

The **Email Groups** tab displays a list of the currently defined email groups. Highlight a group name and click the edit icon (at the far left end of the row) to modify an existing group.

To define a new email group, click the Add icon (+) in the upper-right corner of the table.

Email Groups

|   | Group Name |
|---|------------|
|   | <Default>  |
|   | sales      |



Email Group

Group Name

sales

Email group options specify email notifications will be delivered to a specific group member (or members) during a selected time period.

- **To addresses:** Enter a comma-delimited list of recipient addresses in the To addresses field.
- **Reply to addresses:** Enter a comma-delimited list of recipient addresses in the Reply to addresses field.
- **CC addresses:** Enter a comma-delimited list of addresses that will receive a copy of the email in the CC addresses field.
- **BCC addresses:** Enter a comma-delimited list of addresses that will receive a copy of the email (without the knowledge of other recipients) in the BCC addresses field.
- **From address:** Enter the email address that messages to this group should be sent from in the From address field.
- **Subject prefix:** Enter the email subject prefix to this group in the Subject prefix field.
- **From time/To time(HH:MM:SS):** Use the From time and To time hour selectors to select a time range for a group member (or members). When a notification is sent, the server will evaluate the times specified within the group list and send the message to those members whose group entries include the current time. Provide the From time and To time values in the locale of the PEM client host, and the PEM server will translate the time into other time zones as required.

| To addresses  | From address           | From time                           | To time          |
|---|------------------------|-------------------------------------|------------------|
|   | sales@enterprisedb.com | firstname.lastname@enterprisedb.com | 00:00:0023:59:59 |

Options

To addresses

sales@enterprisedb.com

Reply to addresses

CC addresses

BCC addresses

From address

firstname.lastname@enterprisedb.com

Subject prefix

From time

00:00:00

To time

23:59:59

Use the **Email Group** tab to define an email group and its members:

- Provide a name for the email group in the **Group name** field.

Each row within the email group definition will associate a unique set of email addresses with a specific time period. When an alert is triggered, the server will evaluate the times specified in each row and send the message to those group members whose definitions are associated with the time that the alert triggered.

Click the Add icon (+) in the group members table to open the **Options** tab, and add the member addresses that will receive notifications for the time period specified:

- Enter a comma-delimited list of recipient addresses in the **Reply to addresses** field.
- Enter a comma-delimited list of return addresses in the **Reply to addresses** field.
- Enter a comma-delimited list of addresses that will receive a copy of the email in the **Cc addresses** field.
- Enter a comma-delimited list of addresses that will receive a copy of the email (without the knowledge of other recipients) in the **Bcc addresses** field.
- Enter the email address that messages to this group should be sent from in the **From address** field.
- Provide a comment that will be used as a subject line prefix for any emails sent as part of a notification in the **Subject prefix** field.
- Use the **From time** and **To time** time selectors to specify the time range for notifications to the group member(s) that are identified on this row of the email group dialog. When an alert is triggered, the server will evaluate the times specified in each row and send the message to those group members whose definitions include the current time. Provide the **From time** and **To time** values in the locale of the PEM client host, and the PEM server will translate the time into other time zones as required.

When you've identified the member or members that will receive an email during a specific time period, click the add icon to specify another time period and the email addresses that will be notified during those hours. When you've finished defining the email group, click the save icon.

## Deleting an Email Group

To mark an email group for deletion, highlight the group name in the **Email Groups** table and click the delete icon to the left of the name; the alert will remain in the list, but in red strike-through font.



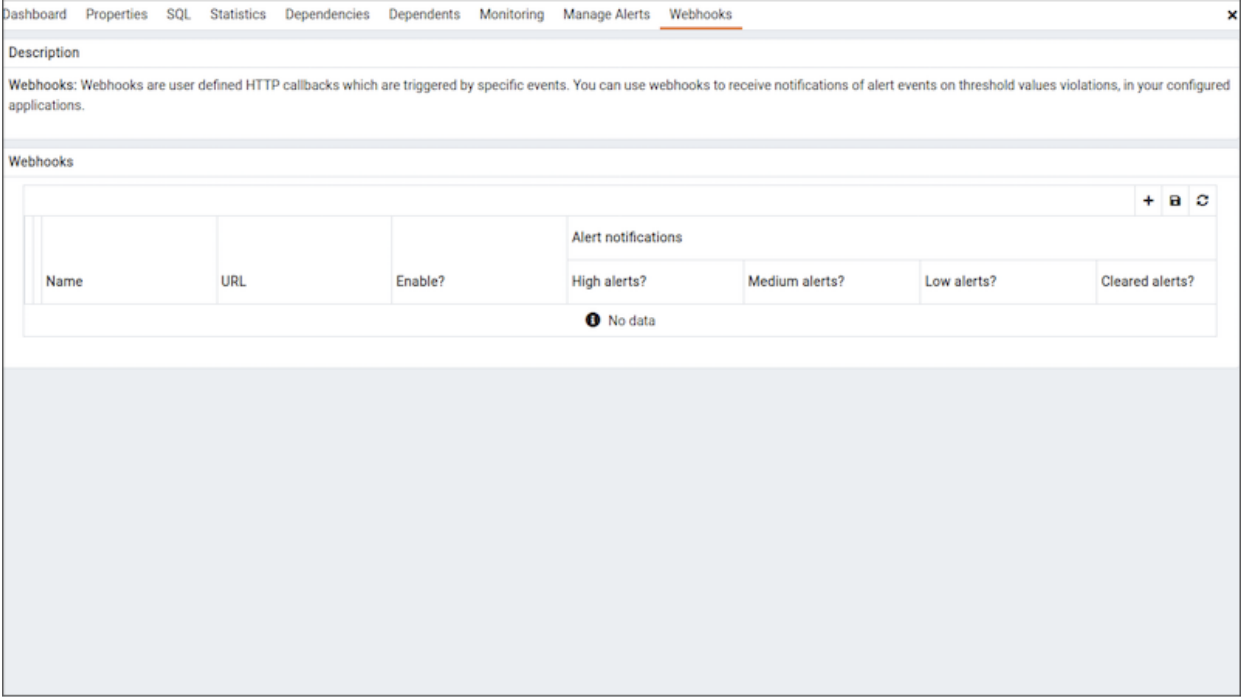
The delete icon acts as a toggle; you can undo the deletion by clicking the delete icon a second time; when you click the save icon, the email group definition will be permanently deleted.

## 34.4.9.6 Webhooks

Postgres Enterprise Manager monitors your system for conditions that require user attention. You can use a webhook to create the endpoints that will receive a notification if current values deviate from threshold values specified in an alert definition. PEM sends a notification to multiple webhook endpoints, or to specific target webhook endpoints based on the events triggered.

Please note that you must configure the PEM Server to use webhooks to receive notification of alert events on threshold value violations in your configured applications.

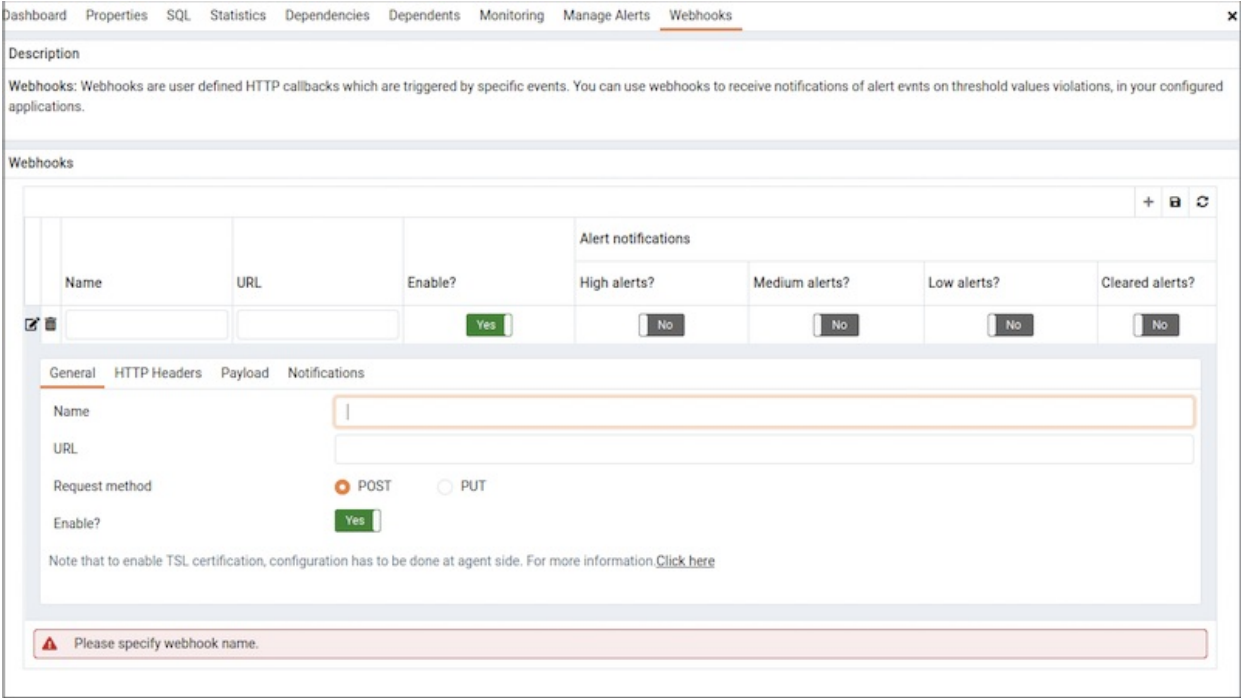
Use the **Webhooks** tab to configure endpoint recipients. To access the **Webhooks** tab, select **Manage Alerts...** from the PEM client's **Management** menu; when the **Manage Alerts** tab opens, select **Webhooks** from the **Quick Links** toolbar.



The **Webhooks** tab displays a list of the currently defined recipient applications as endpoints. Highlight an endpoint and click the edit icon (at the far left end of the row) to modify an existing endpoint.

Creating a Webhook

To define a new webhook, click the **Add** icon (+) in the upper-right corner of the table.



Use the **General** tab to define the basic details of the webhook:

- Provide a name for the webhook in the **Name** field.
- Specify a webhook URL where all the notifications will be delivered in the **URL** field.
- Set the request method type used to make the call in the **Request Method** field i.e. **POST** or **PUT**.

- By default `webhooks` will be enabled; to disable a webhook set `Enable?` to `No` .

Note

The above `Enable?` setting will work only if `enable_webhook` parameter is set to true in `agent.cfg` file. By default, `enable_webhook` parameter is set to true only for the Agent running on the PEM Server Host. For all other Agents running on other hosts, it needs to be set to true manually.

Defining a Webhook SSL configurations

You can define the Webhook SSL parameters in the respective agent configuration file or registry in windows. You can find the list of Webhook SSL parameters in [PEM Agent Configuration Parameters](#) section. If you add or remove any of the agent configuration parameters, you must restart the agent to apply them.

- On Windows systems, PEM registry entries for Webhooks are located in `HKEY_LOCAL_MACHINE\Software\EnterpriseDB\PEM\agent\WEBHOOK`
- On Linux systems, PEM configuration options for Webhooks are stored in the `agent.cfg` file, located (by default) in `/usr/edb/pem/agent/etc`

```
[WEBHOOK/Django]
webhook_ssl_key=<webhook_client_ssl_key_path>
webhook_ssl crt=<webhook_client_ssl_certificate_path>
webhook_ssl ca crt=<webhook_server_ca_certificate_path>
webhook_ssl crt=<crl_file_path_to_validate_webhook_server>
allow_insecure_webhooks=<true|false>
```

EnterpriseDB

languagepack

PEM

agent

WEBHOOK

Django

| Name                  | Type   | Data            |
|-----------------------|--------|-----------------|
| (Default)             | REG_SZ | (value not set) |
| AllowInsecureWebhooks | REG_SZ | true            |
| WebhookSSLCaCrt       | REG_SZ | server_ca.pem   |
| WebhookSSLCrt         | REG_SZ | client.crt      |
| WebhookSSLKey         | REG_SZ | client.key      |

Webhooks

| Name   | URL                                 | Enable? | Alert notifications |                |             |                 |
|--------|-------------------------------------|---------|---------------------|----------------|-------------|-----------------|
|        |                                     |         | High alerts?        | Medium alerts? | Low alerts? | Cleared alerts? |
| Django | http://192.168.1.2:8000/public-w... | Yes     | Yes                 | Yes            | Yes         | Yes             |

General

HTTP Headers

Payload

Notifications

| Key          | Value            |
|--------------|------------------|
| Content-type | application/json |

Use the `HTTP Headers` tab to define the header parameters to pass while calling the webhook endpoints:

- All the values will be specified as a key and value pair.
- Specify a key parameter in the `Key` field and a value in the `Value` field.
- To add multiple `HTTP Headers` , click the `Add` icon (+) in the upper-right corner of the `HTTP Headers` table.
- To delete the `HTTP Headers` , click on `Delete` icon to the left of the `Key` ; the alert will remain in the list, but in strike-through font. Click the

**Save** button to reflect the changes.

- To edit the **HTTP Headers**, click on the **Edit** icon to the left of **Key**.

The screenshot shows the 'Webhooks' configuration interface. At the top, there's a table listing webhooks. Below it, the 'Payload' tab is active, displaying a JSON template for alert notifications. The JSON structure is as follows:

```

1 {
2   "payload": {
3     "version": "1.0",
4     "type": "alert",
5     "source": "e00-pes",
6     "alert": {
7       "AlertID": "%AlertID%",
8       "AlertName": "%AlertName%",
9       "Server/Agent": "%ObjectName%",
10      "ServerID": "null",
11      "ServerName": "%ServerName%",
12      "ServerIP": "%ServerIP%",
13      "ServerPort": "%ServerPort%",
14      "DatabaseName": "%DatabaseName%",
15      "SchemaName": "%SchemaName%",
16      "PackageName": "%PackageName%",
17      "DatabaseObjectName": "%DatabaseObjectName%",
18      "Parameters": "%Parameters%",
19      "AlertInfo": "%AlertInfo%",
20      "Object Type": "%ObjectType%",
21      "Threshold Value": "%ThresholdValue%",
22      "Current Value": "%CurrentValue%",
23      "Current State": "%CurrentState%",
24      "Old State": "%OldState%",
25      "Alert Raised Time": "%AlertRaisedTime%",
26      "Info": "[%CurrentState%]: [%AlertName%] was detected at [%AlertDetected%]"
27    }
28  }
29 }

```

Below the JSON field, there is a 'Test Connection' button and a list of placeholders used in the template:

- %AlertID% - the id of the triggered alert.
- %AlertName% - the name of the triggered alert.
- %ObjectName% - the name of the server or agent on which the alert was triggered.
- %ObjectType% - the type on which alert was generated.
- %ThresholdValue% - the threshold value reached by the metric when the alert triggered.
- %CurrentValue% - the current value of the metric that triggered the alert.
- %CurrentState% - the current state of the alert.
- %OldState% - the previous state of the alert.
- %AlertRaisedTime% - the time that the alert was raised, or the most recent time that the alert state was changed.
- %AgentID% - the id of the agent by which alert was generated.
- %AgentName% - the name of the agent by which alert was generated.
- %ServerID% - the id of the server on which alert was generated.
- %ServerName% - the name of the server on which alert was generated.
- %ServerIP% - the ip or address of the server on which alert was generated.
- %ServerPort% - the port of the server on which alert was generated.
- %DatabaseName% - the name of the database on which alert was generated.
- %SchemaName% - the name of the schema on which alert was generated.
- %PackageName% - the name of the package on which alert was generated.
- %DatabaseObjectName% - the name of the database object name like table name, function name etc on which alert was generated.
- %Parameters% - the list of custom parameters used to generate the alert.
- %AlertInfo% - the detailed database object level information of the alert.

Use the **Payload** tab to define the JSON data to be sent to the endpoint when an alert is triggered:

- Type** specifies data to be sent in format type (i.e. JSON).
- Use **Template** to configure JSON data sent to endpoints. Within the **Template**, you can use placeholders for the following:
  - %AlertID% - the id of the triggered alert.
  - %AlertName% - the name of the triggered alert.
  - %ObjectName% - the name of the server or agent on which the alert was triggered.
  - %ObjectType% - the type on which alert was generated.
  - %ThresholdValue% - the threshold value reached by the metric when the alert triggered.
  - %CurrentValue% - the current value of the metric that triggered the alert.
  - %CurrentState% - the current state of the alert.
  - %OldState% - the previous state of the alert.
  - %AlertRaisedTime% - the time that the alert was raised, or the most recent time that the alert state was changed.
  - %AgentID% - the id of the agent by which alert was generated.
  - %AgentName% - the name of the agent by which alert was generated.
  - %ServerID% - the id of the server on which alert was generated.
  - %ServerName% - the name of the server on which alert was generated.
  - %ServerIP% - the ip or address of the server on which alert was generated.
  - %ServerPort% - the the port of the server on which alert was generated.
  - %DatabaseName% - the name of the database on which alert was generated.
  - %SchemaName% - the name of the schema on which alert was generated.
  - %PackageName% - the name of the package on which alert was generated.
  - %DatabaseObjectName% - the name of the database object name like table name, function name etc on which alert was generated.
  - %Parameters% - the list of custom parameters used to generate the alert.

- `%AlertInfo%` - the detailed database object level information of the alert.
- Click on the `Test Connection` button, to test notification delivery to the mentioned endpoint.

Webhooks

|        |                                     |                                     |  | Alert notifications                 |                                     |                                     |                                     |
|--------|-------------------------------------|-------------------------------------|--|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| Name   | URL                                 | Enable?                             |  | High alerts?                        | Medium alerts?                      | Low alerts?                         | Cleared alerts?                     |
| Django | http://192.168.1.2:8000/public-w... | <input checked="" type="checkbox"/> |  | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |

General HTTP Headers Payload Notifications

Select Yes to enable the alert notifications based on threshold value violations. You can override these notification setting from manage alert tab specific to alert.

All alerts? ☒ Select Yes to enable all the alert notifications.

Alert notifications

High alerts? ☒

Medium alerts? ☒

Low alerts? ☒

Cleared alerts? ☒

Use the `Notifications` tab to specify an alert level for webhook endpoints:

- Set `All alerts` to `Yes` to enable all alert levels to send notifications.
- To instruct PEM to send an notification when a specific alert level is reached, set the slider next to an alert level to `Yes`. Please note that you must set `All alerts` to `No` to configure an individual alert level.

Deleting a Webhook

To mark a webhook for deletion, highlight the webhook name in the `Webhooks` table and click the delete icon to the left of the name; the alert will remain in the list, but in strike-through font.

Webhooks

|        |  |                                     |  | Alert notifications                 |                                     |                                     |                                     |
|--------|--|-------------------------------------|--|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| Name   | URL                                    | Enable?                             |  | High alerts?                        | Medium alerts?                      | Low alerts?                         | Cleared alerts?                     |
| Django | http://192.168.1.2:8000/public-webh... | <input checked="" type="checkbox"/> |  | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Slack  | https://hooks.slack.com/services/T0... | <input checked="" type="checkbox"/> |  | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |

The delete icon acts as a toggle; you can undo the deletion by clicking the delete icon a second time; when you save your work (by clicking the save icon), the webhook definition will be permanently deleted.

34.4.9.7 SNMP MIB Generation

PEM allows alerts to be sent as SNMP traps or notifications to receivers such as network monitoring tools. To enable such tools to understand these notifications, a MIB file may be generated that describes the different alerts and accompanying information that PEM may send. The `pem.generate_alert_mib()` SQL function in the PEM database may be used to generate the MIB file from the alert templates defined in the database. For example:

```
psql.exe -U postgres -d pem -A -t -c "SELECT pem.generate_alert_mib();" > PEM-ALERTING-MIB
```

### 34.4.9.8 SNMP Trap Details

Every SNMP trap send by PEM starts with oid .1.3.6.1.4.1.27645.5444, Significance of each identifier in oid is as follow's.

| Identifier | Meaning   |
|------------|---|
| 1          | ISO, ISO is the group that established the OID standard                         |
| 3          | org, Organization identification schemes registered according to ISO/IEC 6523-2 |
| 6          | dod, United States Department of Defense (DoD)                                  |
| 1          | internet, Communication will be via Internet/network                            |
| 4          | private, This is a device manufactured by a private entity (not goverment)      |
| 1          | enterprise, The device manufacturer is classified as an enterprise              |
| 27645      | PostgreSQL global development group   |
| 5444       | pem   |

#### How OID's are formed?

PEM's SNMP trap has following oid format 1.3.6.1.4.1.27645.5444.<alert\_target\_level\_identifier>.<alert\_identifier>

Following table lists down possible values for `<alert_target_level_identifier>` .

| Identifier | Alert Target Level |
|------------|--------------------|
| 1          | Agent              |
| 2          | Server             |
| 3          | Database           |
| 4          | Schema             |
| 5          | Table              |
| 6          | Index              |
| 7          | Sequence           |
| 8          | Function           |
| 9          | Global             |

`<alert_identifier>` is unique identifier for each alert, which you can find in `snmp_oid` column of `pem.alert_template` table.

For example, snmp\_oid for `Agent Down` alert template is 34, hence trapOID for agent down alert will be 1.3.6.1.4.1.27645.5444.1.34

#### How OID's for binding variables are formed?

Every binding variable oid has following format 1.3.6.1.4.1.27645.5444.10.<binding\_variable\_identifier>, where 10 is identifier for binding variable



Following table lists down possible values for `<binding_variable_identifier>`

| Identifier | Variable Name       |
|------------|---------------------|
| 1          | alertName           |
| 2          | agentID             |
| 3          | serverID            |
| 4          | agentName           |
| 5          | serverName          |
| 6          | databaseName        |
| 7          | schemaName          |
| 8          | objectName          |
| 9          | thresholdvalue      |
| 10         | previousValue       |
| 11         | value               |
| 12         | previousStatus      |
| 13         | status              |
| 14         | recordedTime        |
| 15         | downObjects         |
| 16         | detailedInformation |

For example, 1.3.6.1.4.1.27645.5444.10.1 is oid for binding variable `alertName` .

Details of each snmp traps in `pem.snmp_spool` table. For example,

```
pem=# select * from pem.snmp_spool;
-[ RECORD 1 ]-----+-----
id              | 1
trap_oid        | .1.3.6.1.4.1.27645.5444.1.34
enterprise_oid  | .1.3.6.1.4.1.27645.5444
trap_version    | 2
varbinding_oid  |
.1.3.6.1.4.1.27645.5444.10.1|.1.3.6.1.4.1.27645.5444.10.2|.1.3.6.1.4.1.27645.5444.10.4|.1.3.6.1.4.1.27645.544
10.9|.1.3.6.1.4.1.27645.5444.10.10|.1.3.6.1.4.1.27645.5444.10.11|.1.3.6.1.4.1.27645.5444.10.12|.1.3.6.1.4.1.2
45.5444.10.13|.1.3.6.1.4.1.27645.5444.10.14
varbinding_value | Agent Down||Postgres Enterprise Manager Host|{0.1,0.2,0.3}|0|1|CLEAR|HIGH|2020-06-22
15:51:03.266437+10
sent_status     | s
recorded_time   | 22-JUN-20 15:51:03.266437 +10:00
```

34.4.9.9 Using PEM with Nagios

The PEM server can send a passive alert result to Nagios network-alerting software when an alert is triggered. To instruct the PEM server to notify Nagios of a triggered alert, you must:

- Enable Nagios notification for each alert that will trigger a notification from the PEM server to Nagios. Please note that PEM alerting must be configured before you create the `host.cfg` file and `services.cfg` file.
- Configure Nagios-related behaviors of the PEM server.
- Create the `host.cfg` and `services.cfg` configuration files.
- If necessary, modify the Nagios configuration file and restart the Nagios server.

- Install the PEM Agent on the system where Nagios server is installed and register it with the PEM Server. Set `enable_nagios` configuration to `true` in the `agent.cfg` for that agent, and restart the agent service.

Detailed information about each configuration step is listed below.

After configuring the server to enable Nagios alerting, any triggered alerts will send a passive check result to the Nagios service. The syntax of a passive alert is:

```
[timestamp] PROCESS_SERVICE_CHECK_RESULT; host_name ; service_name ; service_status ;
```

Where:

- `timestamp` is the date and time that the alert was triggered.
- `host_name` is the name of the server or agent.
- `service_name` is the name of the alert.
- `service_status` is the numeric service status value:
  - 0 if the service status is *OK*
  - 1 if the service status is *WARNING*
  - 2 if the service status is *CRITICAL*
  - 3 if the service status is *UNKNOWN*

The PEM server uses the following rules to evaluate the service status:

- If the PEM alert level is `CLEARED`, the warning message will read *OK*
- If the PEM alert level is `LOW`, the warning message will read *WARNING*
- If the `is_nagios_medium_alert_as_critical` flag (specified in the PEM server configuration dialog) is set to `FALSE` and the alert level is `MEDIUM`, the warning message will read *WARNING*
- If the `is_nagios_medium_alert_as_critical` flag (specified in the PEM server configuration dialog) is set to `TRUE` and the alert level is `MEDIUM`, the warning message will read *CRITICAL*
- If the PEM alert level is `HIGH`, the warning message will read *CRITICAL*

## Enabling Nagios Notification for an Alert

The PEM server maintains a unique set of notification properties for each enabled alert. Use the `Notification` tab of the `Manage Alerts` tab to specify that (when triggered), a given alert will send an alert notice to Nagios. To modify the notification properties of an alert, right-click on the name of the object monitored by the alert, and select `Manage Alerts...` from the `Management` menu. When the `Manage Alerts` tab opens, locate the alert, and then click the edit button to the left of the alert name in the `Alerts` list. When the `Manage Alerts` tab opens, select the `Notification` tab.

General
Notification

### Email notification

All alerts?
☐ No

<Default>

Low alerts?
☐ No

<Default>

Medium alerts?
☐ No

<Default>

High alerts?
☐ No

<Default>

To configure notifications for an alert, use the fields in the Email notification box to specify the user or user group that will receive an email notification if the alert is triggered at the specified level. Use the drop-down listbox to select a pre-defined group that will be sent a notification if an alert of the selected level is triggered. Please note that you must configure the PEM Server to use an SMTP server to deliver email before PEM can send email notifications.

### Trap notification

Send trap?
☐ No
SNMP version
v3
Low alert?
☐ No

Medium alert?
☐ No
High alert?
☐ No

Use the Trap notification options to configure trap notifications for this alert. Note that you must configure the PEM Server to send notifications to an SNMP trap/notification receiver before notifications can be sent.

### Nagios notification

Submit passive service check result to Nagios?
☒ Yes

Set "Submit passive service check result to Nagios" to "Yes" to instruct the PEM server to notify Nagios when the alert is triggered or cleared.

### Script execution

Execute script?
☐ No
Execute on alert cleared?
☐ No

Execute script on

☐ PEM Server
☒ Monitored Server

Code

Use the fields in the Script execution box to (optionally) define a script that will be executed if an alert is triggered, and to specify details about the script execution.

- Set the Execute script slider to Yes to instruct PEM to execute the provided script if an alert is triggered.
- Set the Execute on alert cleared slider to Yes to instruct PEM to execute the provided script when the situation that triggered the alert has been resolved.
- Use the selector to indicate if the script should execute on the PEM Server or the Monitored Server.
- Provide the script that PEM should execute in the Code field. You can provide a batch/shell script, or SQL code. Within the script you can use the placeholders to replace the following:
  - %AlertName% - the name of the triggered alert.
  - %ObjectName% - the name of the server or agent on which the alert was triggered.
  - %ThresholdValue% - the threshold value reached by the metric when the alert triggered.
  - %CurrentValue% - the current value of the metric that triggered the alert.
  - %CurrentState% - the current state of the alert.
  - %OldState% - the previous state of the alert.
  - %AlertRaisedTime% - the time that the alert was raised, or the most recent time that the alert state was changed.

To enable Nagios notification, move the slider next to **Submit passive service check result to Nagios** to **Yes**; before exiting the **Manage Alerts** tab, click the save icon to preserve your changes.

## Configuring Nagios-related behavior of the PEM Server

You can use the **Server Configuration** dialog to provide information about your Nagios configuration to the PEM server. To open the **Server Configuration** dialog, select **Server Configuration...** from the PEM client's **Management** menu.

| Server Configuration             |  |                            |
|----------------------------------|--|----------------------------|
|                                  |  | Q Search by parameter name |
| flapping_detection_state_change  | 3  |                            |
| job_failure_notification         | <input type="checkbox"/> False           | t/f                        |
| job_notification_email_group     | default                                  |                            |
| job_retention_time               | 30                                       | days                       |
| job_status_change_notification   | <input type="checkbox"/> False           | t/f                        |
| long_running_transaction_minutes | 5  | minutes                    |
| max_metrics_per_group_chart      | 16                                       |                            |
| nagios_cmd_file_name             | /usr/local/nagios/var/rw/nagios.cmd      |                            |
| nagios_enabled                   | <input checked="" type="checkbox"/> True | t/f                        |
| nagios_medium_alert_as_critical  | <input type="checkbox"/> False           | t/f                        |
| nagios_spool_retention_time      | 7  | days                       |
| probe_log_retention_time         | 30                                       | days                       |
| reminder_notification_interval   | 24                                       | hours                      |
| server_log_retention_time        | 30                                       | days                       |
| show_data_points_on_graph        | <input type="checkbox"/> False           | t/f                        |
| show_data_tab_on_graph           | <input type="checkbox"/> False           | t/f                        |
| show_unmanaged_servers           | <input checked="" type="checkbox"/> True | t/f                        |

?

✕ Cancel
↺ Reset
💾 Save

Four server configuration parameters specify information about your Nagios installation and PEM server behavior related to Nagios:

- Use the `nagios_cmd_file_name` parameter to specify the location of the Nagios pipeline file that will receive passive check alerts from PEM. The default value of this parameter is `/usr/local/nagios/var/rw/nagios.cmd`. The parameter specifies the default file location; if your `nagios.cmd` file resides in an alternate location, specify the file location in the **Value** field.
- Move the slider in the `nagios_enabled` parameter to **Yes** to instruct the PEM server to send passive check alerts to Nagios.
- Use the `nagios_medium_alert_as_critical` slider to specify the warning severity that the PEM server will pass to Nagios if a medium alert is triggered:
  - If the `is_nagios_medium_alert_as_critical` flag is set to **FALSE** and the alert level **MEDIUM**, the warning message will read *WARNING*
  - If the `is_nagios_medium_alert_as_critical` flag is set to **TRUE** and the alert level **MEDIUM**, the warning message will read *CRITICAL*
- Use the `nagios_spool_retention_time` parameter to specify the number of days of notification history that will be stored on the PEM server. The default value is 7 days.

After modifying parameter values, click the save icon to preserve your changes.

## Creating the hosts.cfg and services.cfg File

The `templates.cfg` file (by default, located in `/usr/local/nagios/etc/objects`) specifies the properties of a generic-host and generic-service. The properties specify the parameters used in the `hosts.cfg` and `services.cfg` files.

In most cases (when PEM is installed in a default configuration), you will not be required to modify the `templates.cfg` file before creating the `hosts.cfg` and `services.cfg` files. If necessary, you can modify the `templates.cfg` file to specify alternate values for parameters or to create new templates.

Before modifying the Nagios configuration file, use the following command to create a `hosts.cfg` file that contains information about the PEM hosts that reside on the local system:

```
./psql -U postgres -p 5433 -d pem -A -t -c "select pem.create_nagios_host_config('generic-host') > /usr/local/nagios/etc/objects/hosts.cfg"
```

Then, use the following command to create a `services.cfg` file that contains information about the PEM services that reside on the local system:

```
./psql -U postgres -p 5433 -d pem -A -t -c "select pem.create_nagios_service_config('generic-service') > /usr/local/nagios/etc/objects/services.cfg"
```

If you wish to use a custom `template.cfg` file entry, specify the entry name in place of `generic-host` or `generic-service` in the above commands.

Modifying the Nagios Configuration File

After creating the `host.cfg` and `services.cfg` files, you must specify their location in the Nagios configuration file (by default, `/usr/local/nagios/etc/nagios.cfg`). Modify the configuration file, adding entries that specify the location of the files:

```
cfg_file=/usr/local/etc/objects/hosts.cfg cfg_file=/usr/local/etc/objects/services.cfg
```

You can use the following command to confirm that Nagios is properly configured:

```
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

After confirming that Nagios is configured correctly, restart the Nagios service:

```
/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
```

34.4.10 Using the Manage Charts Tab

You can use the `Manage Charts` tab to access dialogs that allow you to create or modify a custom line chart or table, or import a Capacity Manager template for use in a custom chart. After defining a chart, you can display the chart on a custom dashboard. To open the `Manage Charts` tab, select `Manage Charts...` from the PEM client `Management` menu.

Description

The PEM chart manager allows you create, edit, or delete custom charts. Custom charts are displayed on user-defined custom dashboards. Capacity Manager templates store metric definitions that can be re-used on future charts.

Quick Links

Create New Chart

Import Capacity Manager Template

Help

Custom Charts

Import

|                          | Name                                       | Type       | Level | Metrics Category |
|--------------------------|--|------------|-------|------------------|
| <input type="checkbox"/> | <input checked="" type="checkbox"/> Test-1 | Line Chart | Agent | Alerts           |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> Test-2 | Line Chart | Agent | Alerts           |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> Test-3 | Line Chart | Agent | Audit logs       |

The `Manage Charts` tab provides a `Quick Links` menu that allows you to access dialogs to:

- [Create a New Chart](#) for use on a custom dashboard.
- [Import a Capacity Manager template](#) to use as a template for creating a custom chart.

The **Custom Charts** table displays a list of user-defined charts; when a chart is newly added, the font displays in green. When you add an additional chart or refresh the screen, the name of the chart is displayed in black.

| Custom Charts            |  |            |       |                  | Import |
|--------------------------|--|------------|-------|------------------|--------|
|                          |  |            |       |                  | ⌵ ↺ ⌵  |
| <input type="checkbox"/> | Name                                       | Type       | Level | Metrics Category |        |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> Test-1 | Line Chart | Agent | Alerts           |        |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> Test-2 | Line Chart | Agent | Alerts           |        |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> Test-3 | Line Chart | Agent | Audit logs       |        |

Use the search box in the upper-right hand corner of the **Custom Charts** section to search through your custom charts. Specify a:

- Chart name
- Type
- Level
- Metrics Category

Use icons to the left of a charts name in the **Custom Charts** table to manage a chart:

- Click the edit icon to open the **Chart Configuration** wizard and modify aspects of the chart or table.
- Click the delete icon to delete the selected chart.

### 34.4.10.1 Creating a New Chart

Click the **Create New Chart** icon in the **Quick Links** section of the **Manage Charts** tab to open the **Create Chart** wizard. The **Create Chart** wizard will walk you through the steps required to define a new chart.

Create Chart - Chart Configuration (step 1 of 4)

Follow the steps to create/update the chart:

1

Configure Chart

2

Select Metrics

3

Set Options

4

Set Permissions

Name

Top\_Five\_Table\_Chart

Category

Database Object Activity

Type

Line chart

Table

Description

?

Cancel

Back

Next

Finish

Use the fields on the **Configure Chart** dialog to specify general information about the chart:

- Specify the name of the chart in the **Name** field.
- Use the drop-down listbox in the **Category** field to specify the category in which this chart will be displayed; when adding a custom chart to a custom dashboard, the chart will be displayed for selection in the **Category** specified.
- Use the radio buttons in the **Type** field to specify if the chart will be a **Line chart** or a **Table**.
- Provide a description of the chart in the **Description** field. The description will be displayed to the user viewing the chart (on a custom dashboard) when they click the information icon.

When you've completed the fields on the **Configure Chart** dialog, click **Next** to continue.

**Create Chart - Metric Selection (step 2 of 4)**

Follow the steps to create/update the chart:

- 1 Configure Chart ✓
- 2 **Select Metrics** ➤
- 3 Set Options
- 4 Set Permissions

**Metric level** Database

**Available metrics**

- > Index Size
- > Index Statistics
- > Materialized Vie
- > Materialized Vie
- > Materialized Vie
- > Slony Replicatio
- > SQL/Protect
- > Table Bloat
- > Table Frozen XII
- ▼ **Table Size**
  - Size of Index
  - Table Size (M
  - Total Table S
- > Table Statistics
- ▼ **Top\_Five\_Large\_**
  - table\_size+**
- > xDB Replication

To add a metric to a line chart, locate the metric in the tree control, and double-click the metric

| Metric [Probe]                        | Selection criteria | Order by | Limit |
|---------------------------------------|--------------------|----------|-------|
| table_size+ [ Top_Five_Large_Tables ] | schema_name        | Asc      | 1     |

? Cancel Back Next Finish

Use the fields on the **Select Metrics** dialog to select the metrics that will be displayed on the chart:

- Use the **Metric level** drop-down listbox to specify the level of the PEM hierarchy from which you wish to select metrics. You can specify **Agent**, **Database**, or **Server**. Each level offers access to a unique set of probes and metrics.
- Use the tree control in the **Available metrics** box to select the metrics that will be displayed on the chart.
  - If you are creating a table, you may only select metrics from one probe; each node of the tree control lists the metrics returned by a single probe. Expand a node of the tree control, and check the boxes to the left of a metric name to include that metric data in the table.
  - If you are creating a line chart, expand the nodes of the tree control and double-click each metric that you would like to include in the chart.
- Use the fields in the **Selected metrics** panel to specify how the metric data will be displayed in your chart. The selection panel displays the name of the metric in the (non-modifiable) **Metric [Probe]** column. You can:
  - Click the garbage can icon to delete a metric from the list of selected metrics.
  - Use the drop-down listboxes in the **Selection Criteria** column to specify the order of the data displayed.
  - Use the **Limit** field to specify the number of rows in a table or lines in a chart:

- The maximum number of lines allowed in a chart is 32.
- The maximum number of rows allowed in a table is 100.
- If you are creating a line chart, PEM supports comparisons of cross-hierarchy metrics.
  - Click the compare icon to open a selection box that allows you to select one or more probe-specific attributes (i.e. CPUs, interfaces, databases, etc.) to compare in the chart.
  - Click the copy icon to apply your selections to all of the metrics for the same probe. When the popup opens, click **Yes** to confirm that other selections for the same probe will be overwritten, or **No** to exit the popup without copying the attributes.

When you've completed the fields on the **Select Metrics** dialog, click **Next** to continue.

Use the fields on the **Set Options** dialog to specify display options for your chart:

- Use the **Auto Refresh** field to specify the number of minutes between chart updates - choose a value from 1 to 120. The default auto refresh rate is 2 minutes.

Use fields under the **Line chart options** heading to specify display preferences for a line chart:

- Use the **Points to plot** field to specify the maximum number of points that will be plotted on the chart.
- Use the fields to the right of the **Historical span** label to specify how much historical data should be displayed on the chart:
  - Use the **Day(s)** field to specify the number of days of historical data that should be included on the chart.
  - Use the **Hour(s)** field to specify the number of hours of historical data that should be included on the chart.
  - Use the **Minute(s)** field to specify the number of minutes of historical data that should be included on the chart.
- Use the fields in the **Data extrapolation** box to specify if PEM should generate extrapolated data based on historical data.
  - Click the **No Extrapolation** label to omit extrapolated data from the chart.
  - Click the **Span** label to use the **Days** and **Hours** selectors to specify the period of time spanned by the metrics on the chart.
  - Click the **Threshold** label to use threshold selectors to specify a maximum or minimum value for the chart.



When you've completed the fields on the **Set Options** dialog, click **Next** to continue.

Create Chart - Security (step 4 of 4)

Follow the steps to create/update the chart:

1

Configure Chart ✓

2

Select Metrics ✓

3

Set Options ✓

4

Set Permissions >

Share with all users?

Yes

Access permissions

Specify the user groups that will have access to the new chart.

?

Cancel

Back

Next

Finish

Use the fields on the **Set Permissions** dialog to specify display options for your chart:

- Set the **Share with all** slider to **Yes** to indicate that the chart will be available to all authorized users, or **No** to restrict access to the users or groups specified in the **Access permissions** field.
- Use the **Access permissions** field to select the group or groups that will have access to the chart.

When you've finished defining the chart, click **Finish** to save your edits and add your chart to the list on the **Manage Charts** tab:

Description

The PEM chart manager allows you create, edit, or delete custom charts. Custom charts are displayed on user-defined custom dashboards. Capacity Manager templates store metric definitions that can be re-used on future charts.

Quick Links

Create New Chart

Import Capacity Manager Template

Help

Custom Charts

Search chart by Name, Type, Level or Category

|   | Name                 | Type       | Level    | Metrics Category         |
|---|----------------------|------------|----------|--------------------------|
| ✓ | test1                | Table      | Database | Database Object Activity |
| ✓ | Test1                | Line Chart | Agent    | Alerts                   |
| ✓ | Top_Five_Table_Chart | Line Chart | Database | Database Object Activity |

34.4.10.2 Importing a Capacity Manager Template

Selecting the **Import Capacity Manager Template** from the **Manage Charts** tab's **Quick Links** section opens the **Create Chart**

dialog, allowing you to select from your saved Capacity Manager templates. When the dialog opens, use the `Import capacity template` drop-down listbox to select the template you would like to use for your chart.

Create Chart - Chart Configuration (step 1 of 4)

Follow the steps to create/update the chart:

1

Configure Chart

2

Select Metrics

3

Set Options

4

Set Permissions

Import capacity template

Templates.Cap\_man\_blks\_read\_write

Name

Cap\_man\_blks\_read\_write

Category

Database I/O

Type

☒ Line chart

☐ Table

Description

?

Cancel

Back

Next

Finish

Use the fields on the `Create Chart` dialog to provide information about the chart:

- Specify the name of the chart in the `Name` field.
- Use the drop-down listbox in the `Category` field to specify the category in which this chart will be displayed. When adding a custom chart to a custom dashboard, the chart will be displayed for selection in the `Category` specified.
- Use the radio buttons in the `Type` field to specify if the chart will be a `Line chart` or a `Table`.
- Provide a description of the chart in the `Description` field. The description will be displayed to the user viewing the chart (on a custom dashboard) when they click the information icon.

Click `Next` to continue to the `Select Metrics` window.

Create Chart - Metric Selection (step 2 of 4)

Follow the steps to create/update the chart:

1

Configure Chart ✓

2

Select Metrics >

3

Set Options

4

Set Permissions

Metric level

Capacity Report Chart

| Metrics         | Metric details   |
|-----------------|--|
| Blocks Read+    | Display NameBlocks Read+ (Postgres Enterprise Manager)<br>Probe: IO Analysis<br>Metric: Blocks Read+<br>Host: Postgres Enterprise Manager Host |
| Blocks Read+    | Display NameBlocks Read+ (Postgres Enterprise Manager)<br>Probe: IO Analysis<br>Metric: Blocks Read+<br>Host: Postgres Enterprise Manager Host |
| Blocks Read     | Display NameBlocks Read (Postgres Enterprise Manager)<br>Probe: IO Analysis<br>Metric: Blocks Read<br>Host: Postgres Enterprise Manager Host   |
| Blocks Read     | Display NameBlocks Read (Postgres Enterprise Manager)<br>Probe: IO Analysis<br>Metric: Blocks Read<br>Host: Postgres Enterprise Manager Host   |
| Blocks Written+ | Display NameBlocks Written+ (Postgres Enterprise Manager)<br>Probe: IO Analysis<br>Metric: Blocks Written+                                     |

?

Cancel

Back

Next

Finish

The **Select Metrics** window displays details about the metrics that are used by the template. When you've reviewed the metrics, click **Next** to continue to the **Set Options** window.

### Create Chart - Chart Options (step 3 of 4)

Follow the steps to create/update the chart:

- 1 Configure Chart ✓
- 2 Select Metrics ✓
- 3 Set Options >
- 4 Set Permissions

Auto refresh  Minute(s)  
Please specify the number of minutes between chart updates.

Data extrapolation

☒ Historical days and extrapolated days
 ☐ Historical days and threshold

Historical  Day(s)

Extrapolated  Day(s)

?

✕ Cancel

⏪ Back

Next ⏩

Use the fields on the **Set Options** window to specify display options for your chart:

- Use the **Auto Refresh** field to specify the number of minutes between chart updates - choose a value from 1 to 999. The default auto refresh rate is 2 minutes.
- Use the fields in the **Data extrapolation** box to specify the time period covered by the chart. You can either:
  - click the **Historical days and extrapolated days** label and:
    - specify the number of days of historical data that should be charted in the **Historical** field.
    - specify the number of projected days that should be charted in the **Extrapolated** field.
  - or, click the **Historical days and threshold** label and:
    - provide the number of days of historical data that should be charted in the **Historical** field.
    - use the threshold selection fields to specify the threshold value at which the chart will end.

When you've completed the **Set Options** window, click **Next** to continue.

Create Chart - Security (step 4 of 4)

Follow the steps to create/update the chart:

1

Configure Chart ✓

2

Select Metrics ✓

3

Set Options ✓

4

Set Permissions ➤

Share with all users?

Yes

Access permissions

Specify the user groups that will have access to the new chart.

?

Cancel

Back

Next

Finish

After making any required modifications to the chart definition, click **Finish** to save your edits. PEM will open a popup, confirming that the edits have been saved:

Custom Charts

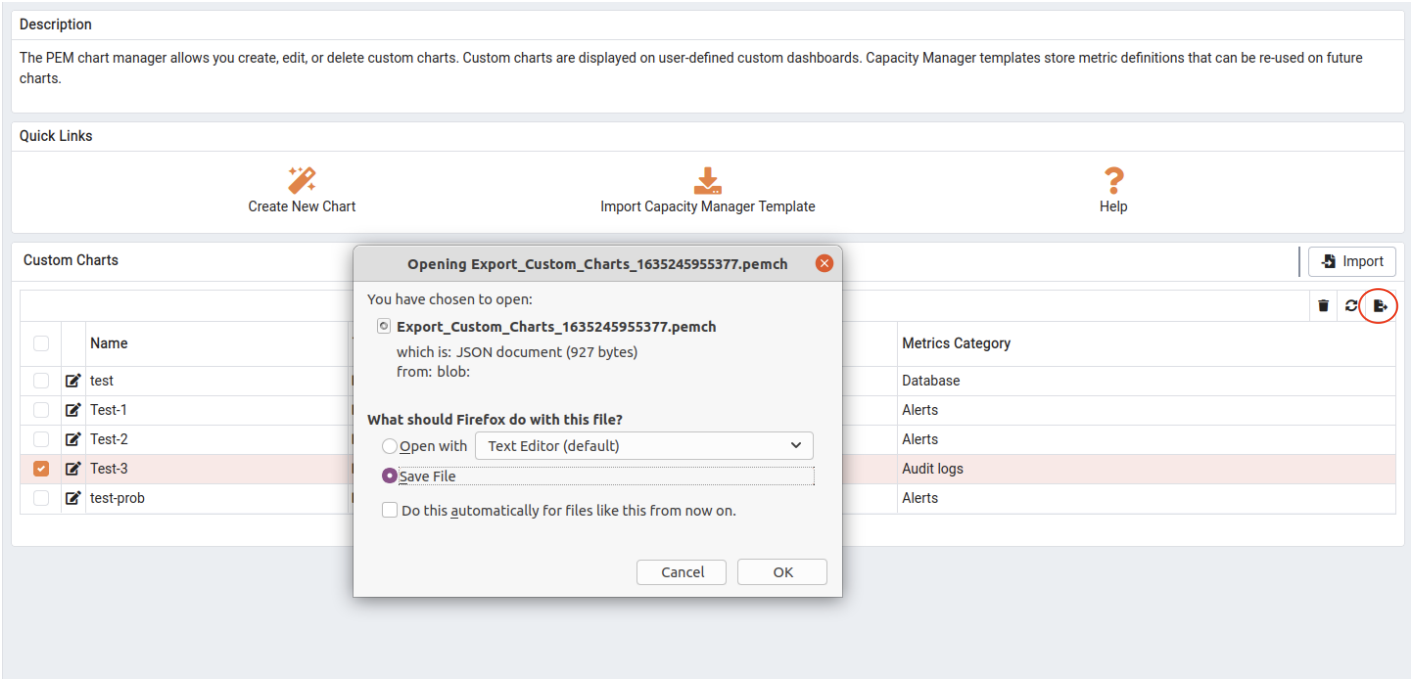
Search chart by Name, Type, Level or Category

|                                     | Name                    | Type           | Level    | Metrics Category         |
|-------------------------------------|-------------------------|----------------|----------|--------------------------|
| <input checked="" type="checkbox"/> | Cap_man_bkls_read_write | Capacity Chart | Database | Database Object Activity |
| <input checked="" type="checkbox"/> | test1                   | Table          | Database | Database Object Activity |
| <input checked="" type="checkbox"/> | Test1                   | Line Chart     | Agent    | Alerts                   |
| <input checked="" type="checkbox"/> | Top_Five_Table_Chart    | Line Chart     | Database | Database Object Activity |

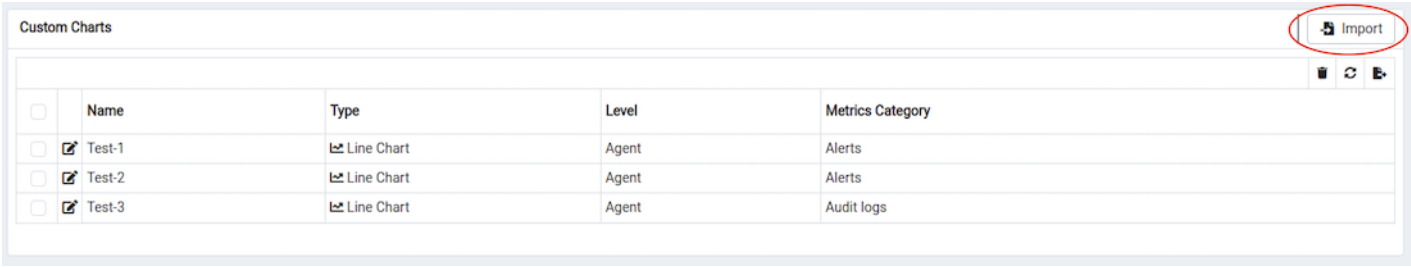
Exporting or Importing a Chart

From PEM 8.3 onwards, you can export or import the charts to another PEM Server.

To **Export** the Chart, select any chart from the **Charts** tab and then select the **Export** icon in the upper-right corner of the table. Select **Save File** option and then select **ok**, it will generate the **JSON** file.



To **Import** the Chart, go to the **Charts** tab and then select the **Import** icon in the upper-right corner.



Click on the **Browse** button to select the **JSON** file with the code to be imported and then click **Import**.



After selecting the file to import, you can select the following checkboxes:

- **skip existing** - If you select this checkbox then it will skip the chart if it already exists.
- **skip existing dependent probe** - The charts are dependent on probes. If you select this checkbox then it will skip the dependent probe if

it already exists.

If both the checkboxes are selected and the chart already exists, then it skips importing the chart with below message:

Import Custom Charts

Select file

Choose File

Export\_Custom\_Charts\_1635156324371.pemch

☒ Skip existing?

☒ Skip existing dependent probes?

Import Summary

| Status | Description                 |
|--------|-----------------------------|
|        | The chart 'test' is skipped |

?

✕ Close

Import

If the `skip existing` checkbox is not selected and `skip dependent probe` is selected and the chart already exists, then it will import the chart successfully as below:

Import Custom Charts

Select file

Choose File

Export\_Custom\_Charts\_1635245405659.pemch

☐ Skip existing?

☒ Skip existing dependent probes?

Import Summary

| Status | Description                                    |
|--------|--|
|        | The chart 'test-prob' is imported successfully |

?

✕ Close

Import

If both the checkboxes are not selected and the chart does not exist, then it will successfully import the chart as below:

Import Custom Charts

Select file

Choose File

Export\_Custom\_Charts\_1635156324371.pemch

☐ Skip existing?

☐ Skip existing dependent probes?

Import Summary

| Status | Description                               |
|--------|---|
| ✓      | The chart 'test' is imported successfully |

?

✕ Close

Import

Deleting an Chart

To delete an chart, select the template name in the charts table, and click the **Delete** icon (located to the upper-right corner of the table). The alert history will persist for the length of time specified on the **History Retention** field in the template definition.

34.4.11 The PEM Manage Dashboards Tab

PEM displays performance statistics through a number of system-defined dashboards; each dashboard contains a series of summary views that contain charts, graphs and tables that display statistics related to the selected object. You can use the Manage Dashboards tab to create and manage custom dashboards that display the information that is most relevant to your system.

Properties SQL Statistics Dependencies Dependents Monitoring Manage Dashboards

Description

You can use the Manage Dashboards tab to create or modify a user-defined dashboard. The custom dashboard may include pre-defined charts, user-defined charts, or a mix of pre-defined and user-defined charts.

Quick Links

Create New Dashboard

Help

Custom Dashboards

Manage Dashboards

Search by Name or Level

|   | Name  | Level    | Description |
|---|-------|----------|-------------|
| ✎ | test1 | Database | test1       |
| ✎ | Test1 | Agent    |             |

To create a custom dashboard, click the **Create New Dashboard** link (located in the **Quick Links** section of the **Manage Dashboards** tab). To modify an existing dashboard, click the edit icon to the left of a dashboard name. The dashboard editor will open, displaying the definition of the dashboard. When you've finished modifying the



dashboard's definition, click the **Save** button to preserve your changes; click **Cancel** to exit without saving your changes. To delete a dashboard, click the delete icon to the left of a dashboard name. A popup will ask you to confirm that you wish to delete the dashboard; click **OK** to delete the selected dashboard.

### 34.4.11.1 Creating a Custom Dashboard

You can use the PEM dashboard editor to create or modify a user-defined dashboard. The custom dashboard may include pre-defined charts, user-defined charts or a mix of pre-defined and user-defined charts. To create a new dashboard, select **Create New Dashboard...** from the **Quick Links** section of the **Manage Dashboards** tab.

Use the fields in the **Configure** section to specify general information about the dashboard:

- Specify a name for the dashboard in the **Name** field. The name specified will also be the title of the dashboard if the title is displayed.
- Use the **Level** drop-down listbox to specify the level of the PEM hierarchy within the PEM client on which the dashboard will be displayed. A dashboard may be accessed via the **Dashboards** menu on a **Global** level, an **Agent** level, the **Server** level or the **Database** level. Each selected level within the list will expose a different set of metrics on which the custom dashboard's charts may be based.
- Provide a description of the dashboard in the **Description** field.

Provide information in the fields in the **Ops dashboard options** box if the dashboard will be used as an Ops dashboard:

- Set the **Ops Dashboard?** field to **Yes** to instruct the server to create a dashboard that is formatted for display on an **Ops monitor**.
- Set the **Show Title?** field to **Yes** to display the dashboard name at the top of the Ops dashboard.
- Use the **Font** drop-down list box to select a custom font style for the title. The selected font style will be displayed in the **Preview** box.
- Use the **Font size** drop-down list box to select a custom font size for the title. The selected font style will be displayed in the **Preview** box.

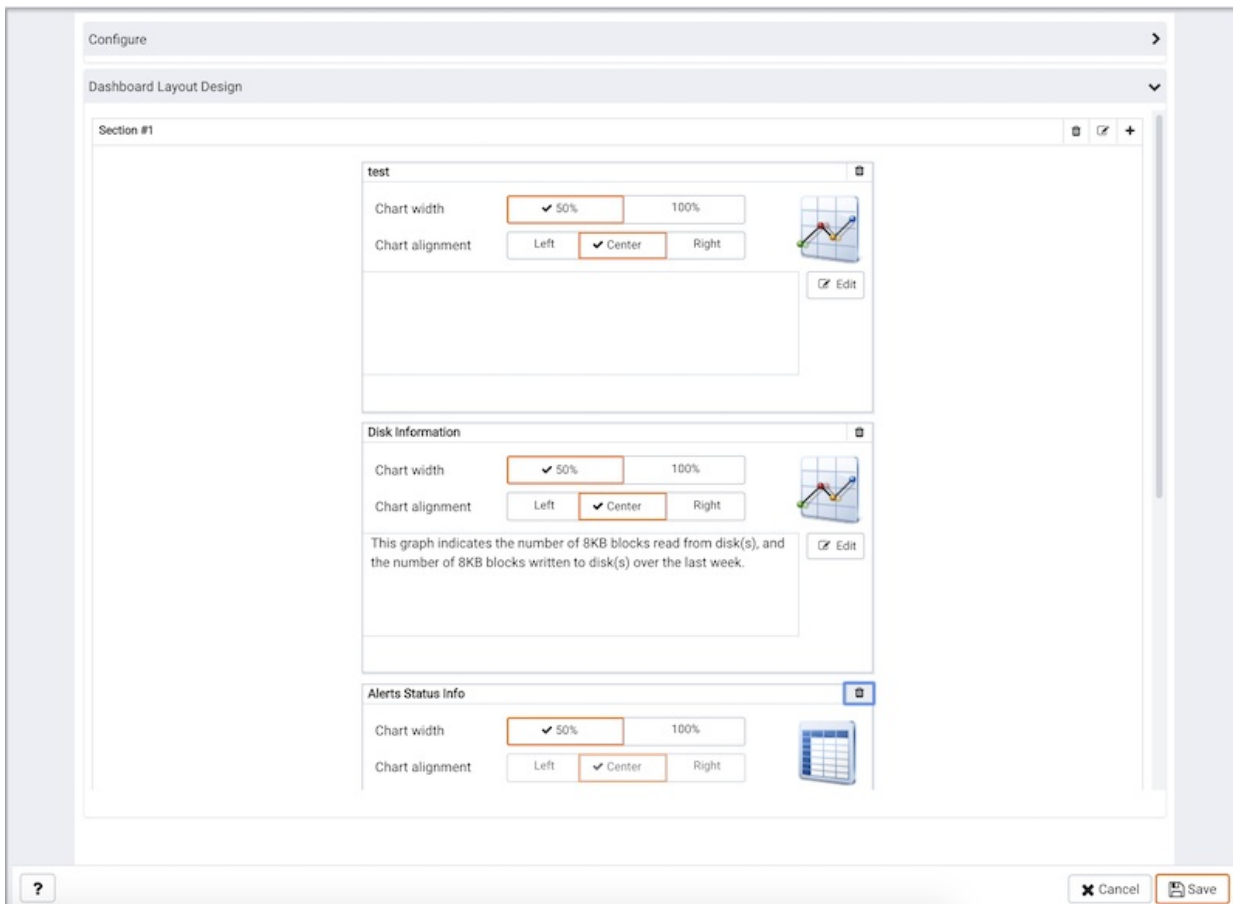
Use the **Permissions** box to specify the users that will be able to view the new dashboard:

- Set the **Share with all** slider to **Yes** to instruct the server to allow all **Teams** to access the dashboard, or set **Share with all** to **No**

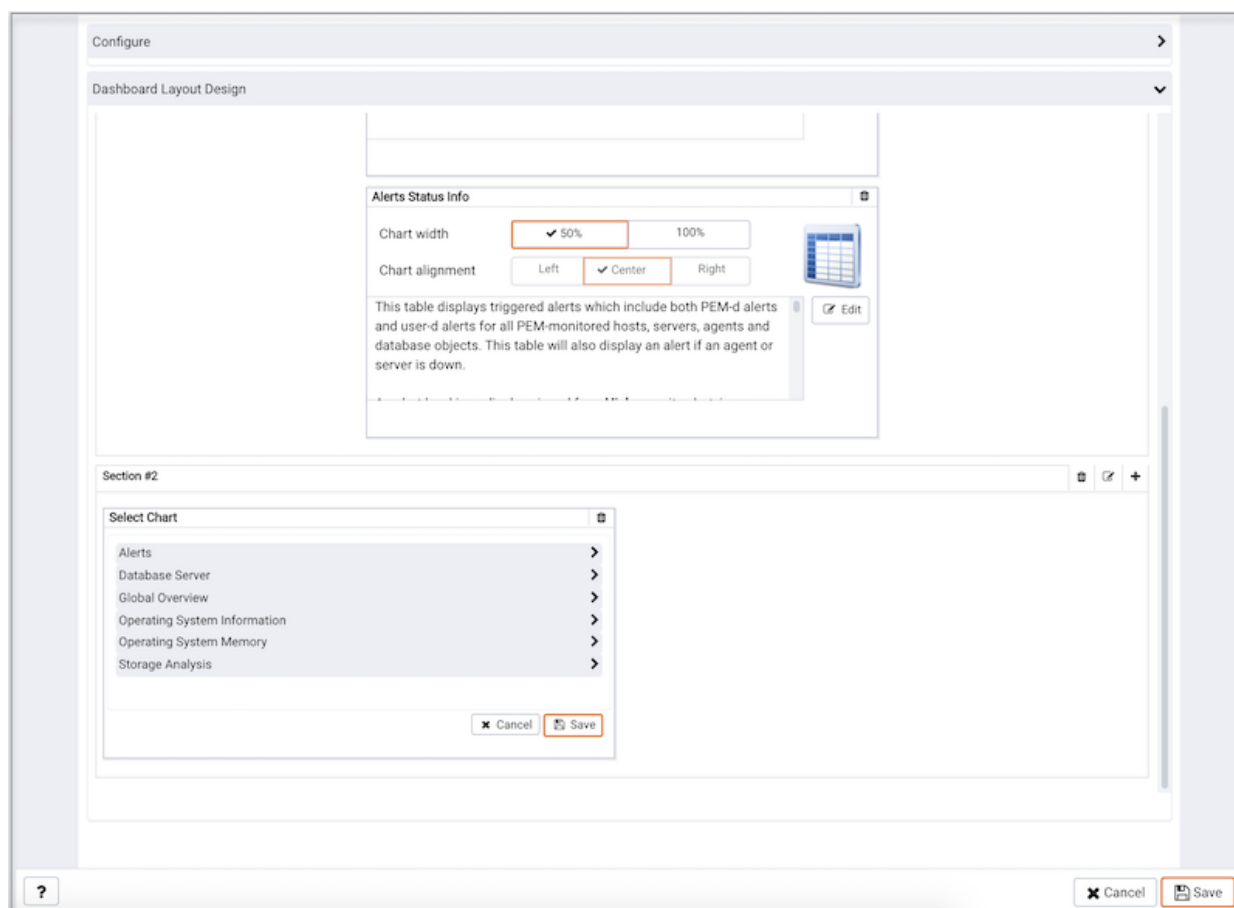
to enable the **Access permissions** field.

- Use the **Access permissions** field to specify which roles can view the new dashboard. Click in the field, and select from the list of users to add a role to the list of users with dashboard access.

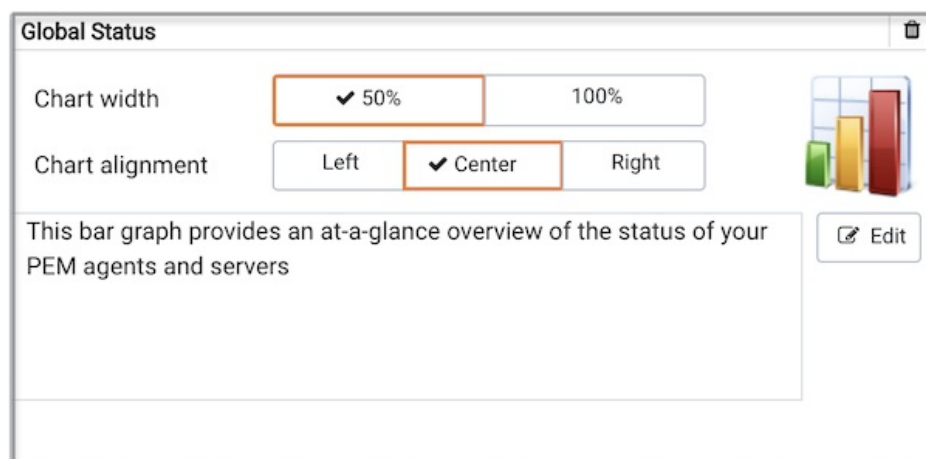
When you've completed the **Configure Dashboard** section, click the arrow in the upper-right corner to close the section, and access the **Dashboard Layout Design** section.



Click the edit icon in a section header to specify a section name; then, click the add icon (+) to add a chart to the section.



Use the arrows to the right of each chart category to display the charts available and select a chart.



Use the chart detail selectors to specify placement details for the chart: \* Use the **Chart width** selector to indicate the width of the chart; select 50% to display the chart in half of the dashboard, or 100% to use the whole dashboard width. \* Use the **Chart alignment** selector to indicate the position of the chart within the section:

- Select **Left** to indicate that the chart should be left-justified.
- Select **Center** to indicate that the chart should be centered.
- Select **Right** to indicate that the chart should be right-justified.

Please note that tables are always displayed centered.

When creating or editing a custom dashboard, you can use drag and drop to re-arrange the charts within a section or to move a chart to a different section.

To add another chart to your dashboard, click the add icon (+) in the section header. When you've finished editing the dashboard, click the **Save** button to

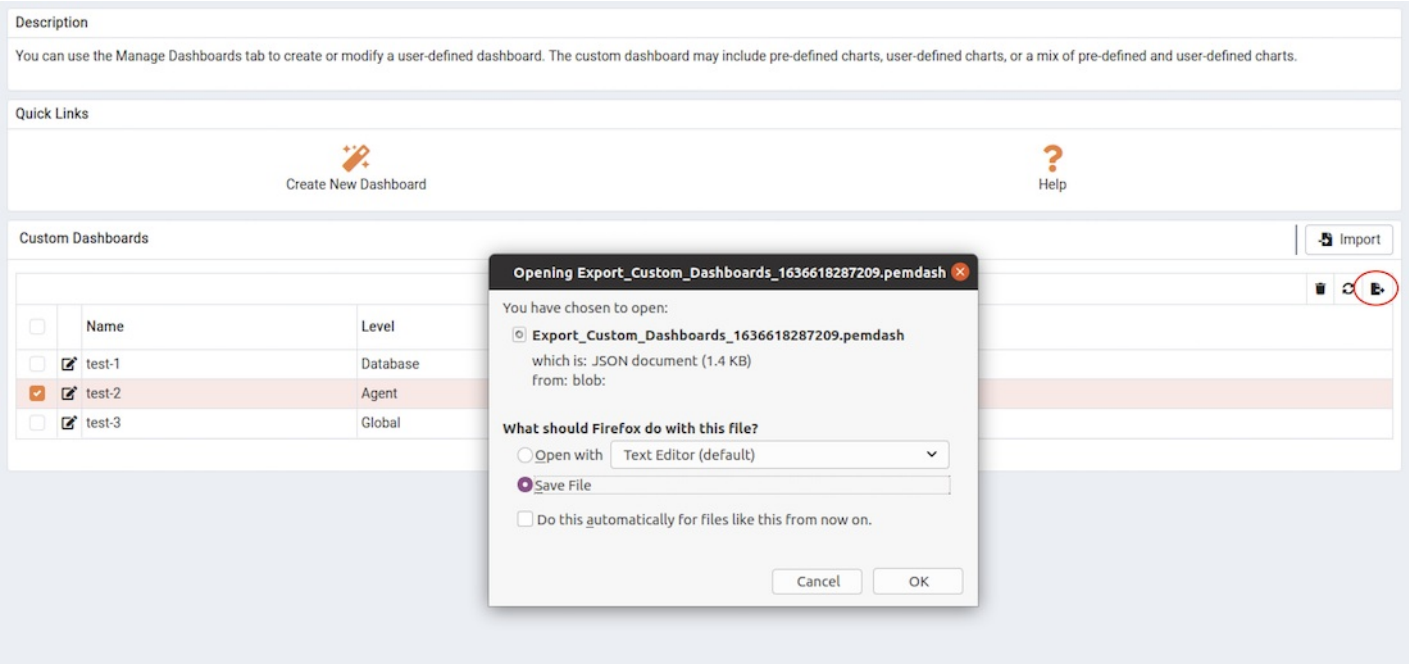
save your edits and exit.

To exit without saving your changes, click the **Cancel** button.

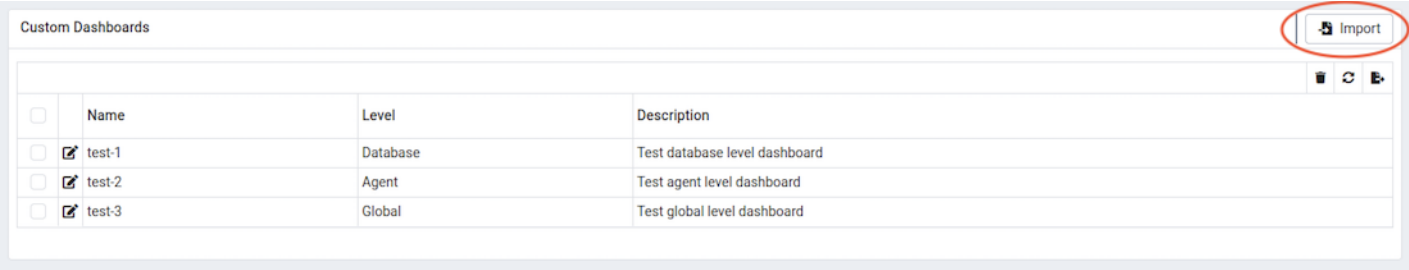
Exporting or Importing a Dashboard

From PEM 8.3 onwards, you can export or import the dashboards to another PEM Server.

To **Export** a dashboard, select any dashboard from the **Dashboards** tab and then select the **Export** icon in the upper-right corner of the table. Select **Save File** option and then select **ok** , it will generate the **JSON** file.



To **Import** a dashboard, go to the **Dashboards** tab and then select the **Import** icon in the upper-right corner.



Click on the **Browse** button to select the **JSON** file with the code to be imported and then click **Import** .

Import Custom Dashboards

Select file

Browse...

Export\_Custom\_Dashbo...635917732083.pemdash

☒ Skip existing?

☒ Skip existing dependent charts?

☒ Skip existing dependent probes?

?

Close

Import

After selecting the file to import, you can select the following checkboxes:

- `skip existing` - If you select this checkbox then it will skip the dashboard if it already exists.
- `skip existing dependent charts` - The dashboards are dependent on charts. If you select this checkbox then it will skip the dependent chart if it already exists.
- `skip existing dependent probes` - The dashboards are dependent on probes. If you select this checkbox then it will skip the dependent probe if it already exists.

If all the checkboxes are selected and the dashboard already exists, then it skips importing the dashboard with below message:

Import Custom Dashboards

Select file

Browse...

Export\_Custom\_Dashboar...1635917732083.pemdash

☒ Skip existing?

☒ Skip existing dependent charts?

☒ Skip existing dependent probes?

Import Summary

| Status       | Description                               |
|--------------|---|
| <div>i</div> | The dashboard 'test-1' is skipped         |
| <div>i</div> | The dashboard 'test-dashboard' is skipped |

?

Close

Import

If the `skip existing`, `skip existing dependent charts` checkboxes are not selected and `skip existing dependent probes` is selected and the dashboard already exists, then it will import the dashboard successfully as below:

Import Custom Dashboards

Select file

Choose File

Export\_Custom\_Dashboards\_1636619894115.pemdash

☒ Skip existing?

☒ Skip existing dependent charts?

☐ Skip existing dependent probes?

Import Summary

| Status | Description   |
|--------|---|
| ✓      | The dashboard 'test-dashboard-probe' is imported successfully |

?

✕ Close

Import

If only `skip existing` checkbox is selected then it skips importing the dashboard with below message:

Import Custom Dashboards

Select file

Choose File

Export\_Custom\_Dashboards\_1636619894115.pemdash

☒ Skip existing?

☐ Skip existing dependent charts?

☐ Skip existing dependent probes?

Import Summary

| Status | Description  |
|--------|--|
| ✕      | The dashboard 'test-dashboard-probe' is failed because it failed to create the chart 'test-prob' due to an error - the probe already exist |

?

✕ Close

Import

If all the checkboxes are not selected and the dashboard does not exist, then it will successfully import the dashboard as below:

Import Custom Dashboards

Select file

Browse...

Export\_Custom\_Dashboards\_1635917732083.pemdash

☐ Skip existing?

☐ Skip existing dependent charts?

☐ Skip existing dependent probes?

Import Summary

| Status | Description   |
|--------|---|
| ✓      | The dashboard 'test-1' is imported successfully         |
| ✓      | The dashboard 'test-dashboard' is imported successfully |

?

✕ Close

Import

Deleting a Dashboard

To delete a dashboard, select the name in the dashboards table, and click the `Delete` icon (located to the upper-right corner of the table).

34.4.11.2 Creating an Ops Dashboard

You can use the PEM `dashboard editor` to create a custom dashboard formatted for display on an Ops monitor. An Ops dashboard displays the specified charts and graphs, while omitting header information and minimizing extra banners, titles, and borders.

To create an Ops dashboard, provide detailed information about the Ops display in the `Ops dashboard options` section of the `Create Dashboard` dialog:

Ops dashboard options

Ops dashboard?

Yes

Show title?

Yes

Font

Arial

Font size

14

Preview

- Set the `Ops Dashboard?` field to `Yes` to instruct the server to create a dashboard that is formatted for display on an Ops monitor.
- Set the `Show Title?` field to `Yes` to display the dashboard name at the top of the Ops dashboard.
- Use the `Font` drop-down list box to select a custom font style for the title. The selected font style will be displayed in the `Preview` box.
- Use the `Font size` drop-down list box to select a custom font size for the title. The selected font style will be displayed in the `Preview` box.

After adding charts and tables to the Ops dashboard, click the `Save` button to save your work. You can then access the dashboard by navigating through the `Dashboards` menu of the hierarchy level specified in the `Level` field on the `New Dashboard` dialog.

34.4.12 The Manage Probes Tab

A [probe](#) is a scheduled task that returns a set of performance metrics about a specific monitored server, database, operating system or agent. You can use the Manage Probes tab to override the default configuration and customize the behavior of each probe. To open the Manage Probes tab, select [Manage Probes...](#) from the [Management](#) menu.

PropertiesSQLStatisticsDependenciesDependentsMonitoringManage Probes

Description

Manage Custom Probes: PEM uses probes to retrieve statistics from a monitored server, database, operating system or agent. You can view, reconfigure, delete, or create your own custom probes.  
Copy Probes: PEM allows copying of probes from any chosen object recursively down through the object hierarchy. Click on Copy Probes to quickly copy the displayed probe configuration to a selected target.

Quick Links

Manage Custom Probes

Copy Probes

Help

Probes

| Probe name                               | Execution Frequency |         |         | Enabled?       |                | Data Retention |      |
|--|---------------------|---------|---------|----------------|----------------|----------------|------|
|  | Default?            | Minutes | Seconds | Default?       | Probe Enable?  | Default?       | Days |
| Background Writer Statistics             | <div>Yes</div>      | 5       | 0       | <div>Yes</div> | <div>Yes</div> | <div>Yes</div> | 180  |
| Blocked Session Information              | <div>Yes</div>      | 5       | 0       | <div>Yes</div> | <div>Yes</div> | <div>Yes</div> | 180  |
| Data and Log File Analysis               | <div>Yes</div>      | 0       | 10      | <div>Yes</div> | <div>Yes</div> | <div>Yes</div> | 180  |
| Database Frozen XID                      | <div>Yes</div>      | 720     | 0       | <div>Yes</div> | <div>Yes</div> | <div>Yes</div> | 180  |
| Database Size                            | <div>Yes</div>      | 30      | 0       | <div>Yes</div> | <div>Yes</div> | <div>Yes</div> | 180  |
| Database Statistics                      | <div>Yes</div>      | 30      | 0       | <div>Yes</div> | <div>Yes</div> | <div>Yes</div> | 90   |
| Failover Manager Cluster Info            | <div>Yes</div>      | 5       | 0       | <div>Yes</div> | <div>No</div>  | <div>Yes</div> | 7    |
| Failover Manager Node Status             | <div>Yes</div>      | 5       | 0       | <div>Yes</div> | <div>No</div>  | <div>Yes</div> | 7    |
| Lock Information                         | <div>Yes</div>      | 5       | 0       | <div>Yes</div> | <div>Yes</div> | <div>Yes</div> | 180  |
| Number of Prepared Transactions          | <div>Yes</div>      | 5       | 0       | <div>Yes</div> | <div>Yes</div> | <div>Yes</div> | 180  |
| Number of WAL Files                      | <div>Yes</div>      | 5       | 0       | <div>Yes</div> | <div>Yes</div> | <div>Yes</div> | 180  |
| Object Catalog: Database                 | <div>Yes</div>      | 5       | 0       | <div>Yes</div> | <div>Yes</div> | <div>Yes</div> | 180  |
| Object Catalog: Tablespace               | <div>Yes</div>      | 5       | 0       | <div>Yes</div> | <div>Yes</div> | <div>Yes</div> | 180  |
| PG HBA Conf                              | <div>Yes</div>      | 30      | 0       | <div>Yes</div> | <div>Yes</div> | <div>Yes</div> | 180  |
| Server Information                       | <div>Yes</div>      | 5       | 0       | <div>Yes</div> | <div>Yes</div> | <div>Yes</div> | 180  |
| Server log Configuration                 | <div>Yes</div>      | 0       | 10      | <div>Yes</div> | <div>Yes</div> | <div>Yes</div> | 180  |
| Session Information                      | <div>Yes</div>      | 5       | 0       | <div>Yes</div> | <div>Yes</div> | <div>Yes</div> | 180  |
| Settings                                 | <div>Yes</div>      | 5       | 0       | <div>Yes</div> | <div>Yes</div> | <div>Yes</div> | 180  |
| Streaming Replication                    | <div>Yes</div>      | 5       | 0       | <div>Yes</div> | <div>No</div>  | <div>Yes</div> | 180  |
| Streaming Replication Database Conflicts | <div>Yes</div>      | 5       | 0       | <div>Yes</div> | <div>No</div>  | <div>Yes</div> | 180  |
| Streaming Replication Lag Time           | <div>Yes</div>      | 5       | 0       | <div>Yes</div> | <div>No</div>  | <div>Yes</div> | 180  |
| Tablespace Size                          | <div>Yes</div>      | 30      | 0       | <div>Yes</div> | <div>Yes</div> | <div>Yes</div> | 180  |
| User Information                         | <div>Yes</div>      | 30      | 0       | <div>Yes</div> | <div>Yes</div> | <div>Yes</div> | 180  |
| WAL Archive Status                       | <div>Yes</div>      | 30      | 0       | <div>Yes</div> | <div>No</div>  | <div>Yes</div> | 180  |

The [Manage Probes](#) tab provides a set of [Quick Links](#) that you can use to create and manage probes:

- Click the [Manage Custom Probes](#) icon to open the [Custom Probes](#) tab and create or modify a custom probe.
- Click the [Copy Probes](#) icon to open the [Copy Probe](#) dialog, and copy the probe configurations from the currently selected object to one or more monitored objects.

A probe monitors a unique set of metrics for each specific object type (server, database, database object, or agent); select the name of an object in the tree control to review the probes for that object.



To modify the properties associated with a probe, highlight the name of a probe, and customize the settings that are displayed in the **Probes** table:

- Move the **Default** switch in the **Execution Frequency** columns to **No** to enable the **Minutes** and **Seconds** selectors, and specify a non-default value for the length of time between executions of the probe.
- Move the **Default** switch in the **Enabled?** column to **No** to change the state of the probe, and indicate if the probe is active or not active.
- Move the **Default** switch in the **Data Retention** column to **No** to enable the **Day(s)** field and specify the number of days that information gathered by the probe is stored on the PEM server.

The **Manage Probes** tab may also display information about probes that cannot be modified from the current node; if a probe cannot be modified from the current dialog, the switches for that probe are disabled. Generally, a disabled probe can be modified from a node that is higher in the hierarchy of the PEM client tree control. Select another object in the tree control to change which probes are displayed or enabled on the **Manage Probes** tab.

### 34.4.12.1 Custom Probes

Use the **Custom Probes** tab to create a new probe or modify an existing probe. After creating or modifying a probe, you can incorporate the data gathered by custom probes into existing or new charts or graphs. To open the **Custom Probes** tab, select the **Manage Custom Probes** icon from the **Quick Links** section of the **Manage Probes** tab.

Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Monitoring
Manage Alerts
Manage Probes
Probes

### Description

**System Probes:** System probes are the built-in probes provided by PEM and are part of the PEM schema. These probes are differentiated in the Probes list by a grey background. You may only modify the Enabled, Interval and Data retention fields of a system probe.

**Probes:** Custom probes are those probes created by users. You can modify the Enabled, Interval and Data retention fields in the General tab, the Unit and Graphable fields of each column on the Columns tab, the code provided in the Code tab, and the code definition on the Alternate Code tab of a user-defined probe if the Applies to all database server versions? field for that probe is set to No.

You may delete only user-defined probes. When you delete a probe, the probe is marked for deletion and will be deleted later (when custom probes are purged). During the deletion the probe definition is deleted and any corresponding tables are dropped from the pemdata and pemhistory schemas.

### Probes

Show System Probes?
No
Import

|                          |                       |                   |             | Execution frequency |         |                |                |
|--------------------------|-----------------------|-------------------|-------------|---------------------|---------|----------------|----------------|
|                          | Probe name            | Collection method | Target type | Minutes             | Seconds | Probe enabled? | Data retention |
| <input type="checkbox"/> | Top_Five_Large_Tables | SQL               | Table       | 1                   | 0       | Yes            | 1              |

Use the **Show system probes?** switch to display the system probes on the **Custom Probes** tab.

To modify an existing probe, click the **Edit** icon located to the left of a probe name.

## Defining a New Probe

To create a new probe, click the **Add** icon in the upper-right corner of the tab.

Probes Show System Probes? ☐ No Import

|                                     | Probe name | Collection method | Target type | Execution frequency |         | Probe enabled?                          | Data retention |
|-------------------------------------|------------|-------------------|-------------|---------------------|---------|---|----------------|
|                                     |            |                   |             | Minutes             | Seconds |   |                |
| <input checked="" type="checkbox"/> |            | SQL               | Server      | 5                   | 0       | <input checked="" type="checkbox"/> Yes | 1              |

General Columns Code Alternate Code

**Probe name** ⚠

**Collection method**   
 Use the Collection method field to specify the probe type. Use the drop-down to select:  
 • SQL (the probe will gather information via a SQL statement)  
 • WMI (the probe will gather information via a Windows Management Instrumentation extension)  
 • Batch/Shell Script (the probe will use a command script or shell script to gather information). Please note that batch probes are platform specific. If you specify a collection method of Batch, you must specify a platform type in the Platform field.

**Target type**   
 Use the Target type drop-down to select the object type that the probe will monitor.

**Execution frequency**  
 Minutes  Seconds

**Probe enabled?** ☒ Yes  
 Use the Enabled? switch to specify if the probe is enabled by default. Specify Yes to enable the probe by default, or No to specify that the probe is disabled by default.

**Data retention**   
 Use the Data retention field to specify the number of days that gathered information will be retained in the probe's history table.

**Discard from history?** ☐ No  
 Use the Discard from history field to specify if the server should create a history table for the probe. Select Yes to discard probe history, or No to retain the probe history in a table.

**Platform**   
 Use the Platform drop-down to specify the type of platform that the probe will monitor. This field is enabled only when the Collection method is Batch/Shell Script.

⚠ Please specify Probe name

Use the fields on the **General** tab to modify the definition of an existing probe or to specify the properties of a new probe.

- Use the **Probe name** field to provide a name for a new probe.
- Use the **Collection method** field to specify the probe type. Use the drop-down listbox to select from:
  - **SQL** (the probe will gather information via a SQL statement)
  - **WMI** (the probe will gather information via a Windows Management Instrumentation extension)
  - **Batch/Shell Script** (the probe will use a command-script or shell-script to gather information).

Before creating a batch probe on a Linux system, you must modify the **agent.cfg** file, setting the **allow\_batch\_probes** parameter equal to **true** and restart the PEM agent. The **agent.cfg** file is located in **/opt/PEM/agent/etc**.

On Windows systems, agent settings are stored in the registry. Before creating a batch probe, modify the registry entry for the **AllowBatchProbes** registry entry and restart the PEM agent. PEM registry entries are located in **HKEY\_LOCAL\_MACHINE\Software\EnterpriseDB\PEM\agent**.

Please note that batch probes are platform-specific. If you specify a collection method of **Batch**, you must specify a platform type in the **Platform** field.

To invoke a script on a Linux system, you must modify the entry for **batch\_script\_user** parameter of agent.cfg file and specify the user that should be used to run the script. You can either specify a non-root user or root for this parameter. If you do not specify a user, or the specified user does not exist, then the script will not be executed. Restart the agent after modifying the file. If pemagent is being run by a non-root user then the value of **batch\_script\_user** will be ignored and the script will be executed by the same non-root user that is being used for running the pemagent.

- Use the **Target type** drop-down listbox to select the object type that the probe will monitor. **Target type** is disabled if **Collection method** is **WMI**.
- Use the **Minutes** and **Seconds** selectors to specify how often the probe will collect data.
- Use the **Probe enable?** switch to specify if the probe is enabled. Specify **Yes** to enable the probe, or **No** to specify that the probe is disabled.
- Set the **Data retention** switch to **Yes** to specify the number of days that gathered information will be retained in the probe's history table.
- Use the switch next to **Discard from history** to specify if the server should create a history table for the probe. Select **Yes** to discard probe history, or **No** to retain the probe history in a table.
- Use the **Platform** drop-down listbox to specify the type of platform that the probe will monitor. This field is enabled only when the **Collection method** is **Batch**.

The screenshot displays the 'Probes' configuration interface. At the top, there's a table listing probes with columns: Probe name, Collection method, Target type, Execution frequency (Minutes, Seconds), Probe enabled?, and Data retention. Below this is a detailed configuration panel for a selected probe, with tabs for General, Columns, Code, and Alternate Code. The 'General' tab is active, showing fields for Name, Internal name, Column type, Data type, Unit, Graphable?, Is PIT?, and Calculate PIT?. The Name field is empty and has a red error message 'Please specify column name'. The Internal name field is disabled. The Column type is set to 'Non key'. The Data type is set to 'numeric'. The Unit field is empty. The Graphable? switch is set to 'Yes'. The Is PIT? switch is set to 'No'. The Calculate PIT? switch is set to 'No'. There are also red error messages at the bottom: 'Please specify Probe name'.

Use the **Columns** tab to define the columns in which the probe data will be stored. Navigate to the Columns tab, and click the **Add** button (in the upper-right corner) to define a new column.

- Provide a name for the column in the **Name** field.
- The **Internal name** field is not enabled for user-defined probes.
- Use the **Column type** drop-down listbox to specify if the column is a **Key** column (a primary key) or a **Non key** column. Non-key columns are generally metric items (values that can be graphed).
- Use the **Data type** drop-down listbox to specify the type of data that will be stored in the column.
- Use the **Unit** field to specify the unit of measure that applies to the metric stored in the column. This unit is displayed on the Y-Axis of a custom chart or a Capacity Manager chart. This is an optional field.

- Use the **Graphable** switch to specify if the defined metric may be graphed, and that the probe should be accessible from the Capacity Manager or Manage Charts dialogs.
- Use the **Is PIT** switch to specify if the metric is stored by point-in-time (by default).

'Point-in-time' metrics are those metrics that change (increase or decrease) at any given point of time. For example, database size is a point-in-time metric; at any given point-in-time, the size of the database is fluctuating. Metrics that are not point-in-time (also referred to as cumulative metrics) are metrics whose size always increases over time. For example, Blocks Read and Tuples Read are cumulative metrics; the value stays the same or increases.

- Use the **Calculate PIT** switch to specify that the server should calculate a point-in-time value for the metric data. **Calculate PIT** is disabled if **Is PIT** is **Yes**.

PEM allows you to store point-in time-values of cumulative metrics as well. PEM subtracts the last collected value of a cumulative metric from the current value, and stores the difference as a point-in-time value.

The screenshot shows the 'Probes' configuration window. At the top, there's a 'Show System Probes?' toggle set to 'No' and an 'Import' button. Below is a table with columns: Probe name, Collection method, Target type, Execution frequency (Minutes, Seconds), Probe enabled?, and Data retention. A single probe is listed with 'SQL' as the collection method, 'Server' as the target type, 5 minutes execution frequency, and 'Yes' for 'Probe enabled?'. Below the table, there are tabs for 'General', 'Columns', 'Code', and 'Alternate Code'. The 'Code' tab is active, showing a large text area for the SQL SELECT statement. A red error message at the bottom says 'Please specify Probe name'.

Use the **Code** tab to specify the default code that will be executed by the probe.

- If the probe is a SQL probe, you must specify the SQL SELECT statement invoked by the probe on the **Code** tab. The column names returned by the query must match the **Internal Name** specified on the **Column** tab. The number of columns returned by the query, as well as the column name, datatype, etc. must match the information specified on the **Columns** tab.
- If the probe is a Batch probe, you must specify the shell or .bat script that will be invoked when the probe runs. The output of the script should be as follows:
  - The first line must contain the names of the columns provided on the **Columns** tab. Each column name should be separated by a tab (t) character.
  - From the second line onwards, each line should contain the data for each column, separated by a tab character.
  - If a specified column is defined as key column, make sure the script does not produce duplicate data for that column across lines of output.
  - The number of columns specified in the **Columns** tab and their names, data type, etc. should match with the output of the script output.
- If the probe is a WMI probe, you must specify the WMI query as a SELECT WMI query. The column name referenced in the SELECT statement should be same as the name of the corresponding column specified on the **Column** tab. The column names returned by the query must match the **Internal Name** specified on the **Column** tab. The number of columns returned by the query, as well as the column name, datatype, etc. must match the information specified on the **Columns** tab.

Use the **Alternate Code** tab to provide code that will be invoked if the probe fires on a specific version of the server. To provide version-specific code, move the **Applies to all database server versions?** switch to **No**, and click the **Add** button. Then, use the **Database version(s)** drop-down listbox to select the version to which the code will apply. After selecting the version, click the **Edit** button (to the left of the version name) to provide the code that will execute when the probe fires.

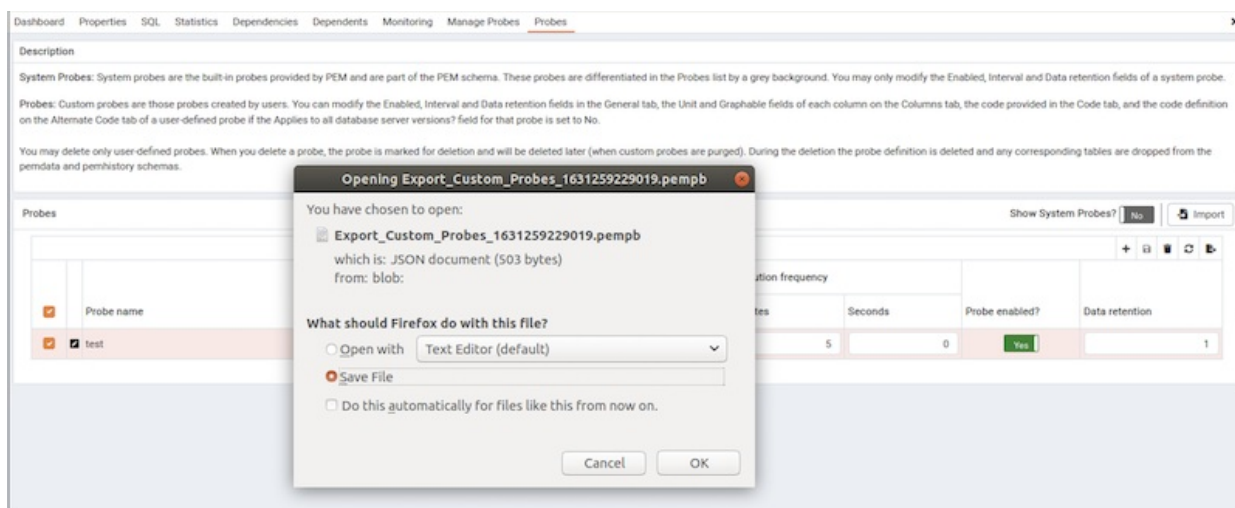
If you select a database version, and leave the **Probe code** column blank, PEM will invoke the code specified on the **Code** tab when the probe executes on a server that matches that version.

When you've finished defining the probe, click the **Save** icon (in the corner of the **Custom Probes** tab) to save the definition, and make the probe data available for use on custom charts and graphs.

## Exporting and Importing Probes

From PEM 8.2 onwards, you can export and import the probes to another PEM Server.

To **Export** the probe, select any probe/s from the **Manage Custom Probes** tab and then select the **Export** icon in the upper-right corner of the table. Select **Save File** option and then select **ok**, it will generate the **JSON** file.



To **Import** the probe, go to the **Manage Custom Probes** tab and then select the **Import** icon in the upper-right corner of the table.

Dashboard Properties SQL Statistics Dependencies Dependents Monitoring Manage Probes **Probes**

**Description**

**System Probes:** System probes are the built-in probes provided by PEM and are part of the PEM schema. These probes are differentiated in the Probes list by a grey background. You may only modify the Enabled, Interval and Data retention fields of a system probe.

**Probes:** Custom probes are those probes created by users. You can modify the Enabled, Interval and Data retention fields in the General tab, the Unit and Graphable fields of each column on the Columns tab, the code provided in the Code tab, and the code definition on the Alternate Code tab of a user-defined probe if the Applies to all database server versions? field for that probe is set to No.

You may delete only user-defined probes. When you delete a probe, the probe is marked for deletion and will be deleted later (when custom probes are purged). During the deletion the probe definition is deleted and any corresponding tables are dropped from the pemdata and pemhistory schemas.

**Probes** Show System Probes?

|                          | Probe name            | Collection method | Target type | Execution frequency |         | Probe enabled? | Data retention |
|--------------------------|-----------------------|-------------------|-------------|---------------------|---------|----------------|----------------|
|                          |                       |                   |             | Minutes             | Seconds |                |                |
| <input type="checkbox"/> | Top_Five_Large_Tables | SQL               | Table       | 1                   | 0       | No             | 1              |

Click on the **Browse** button to select the **JSON** file with the probe code to be imported and then click **Import**.

**Import Custom Probes**

Select file  Export\_Custom\_Probes\_1629814441988.json

☒ Skip existing?

After selecting the file to import you can select the **skip existing** checkbox. If selected then it will skip the probe if it already exists.

If the checkbox is selected and the probe already exists, then it skips importing the probe with the message as below:

**Import Custom Probes**

Select file  Export\_Custom\_Probes\_1629814441988.json

☒ Skip existing?

**Import Summary**

| Status | Description                                  |
|--------|--|
|        | The probe 'Top_Five_Large_Tables' is skipped |

If the checkbox is not selected and the probe already exists, then it does not import the probe and throws the below error:

**Import Custom Probes**

Select file  Export\_Custom\_Probes\_1629814441988.json

☐ Skip existing?

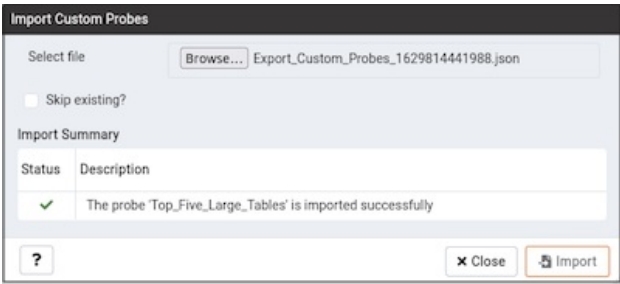
**Import Summary**

| Status | Description   |
|--------|---|
|        | The probe 'Top_Five_Large_Tables' is failed because the probe already exist |

#### Note

**Import** cannot overwrite the existing probe as it may be configured to retain historical data as per the configured retention policy.

If the checkbox is not selected and probe does not exist but the corresponding table in the **pem** schema exists, then it imports the probe successfully using the same table.



**Note**

It is possible that probe is deleted and not listed on **Manage Custom Probe** tab, but still the table holding the data of that probe exists in the **pem** schema.

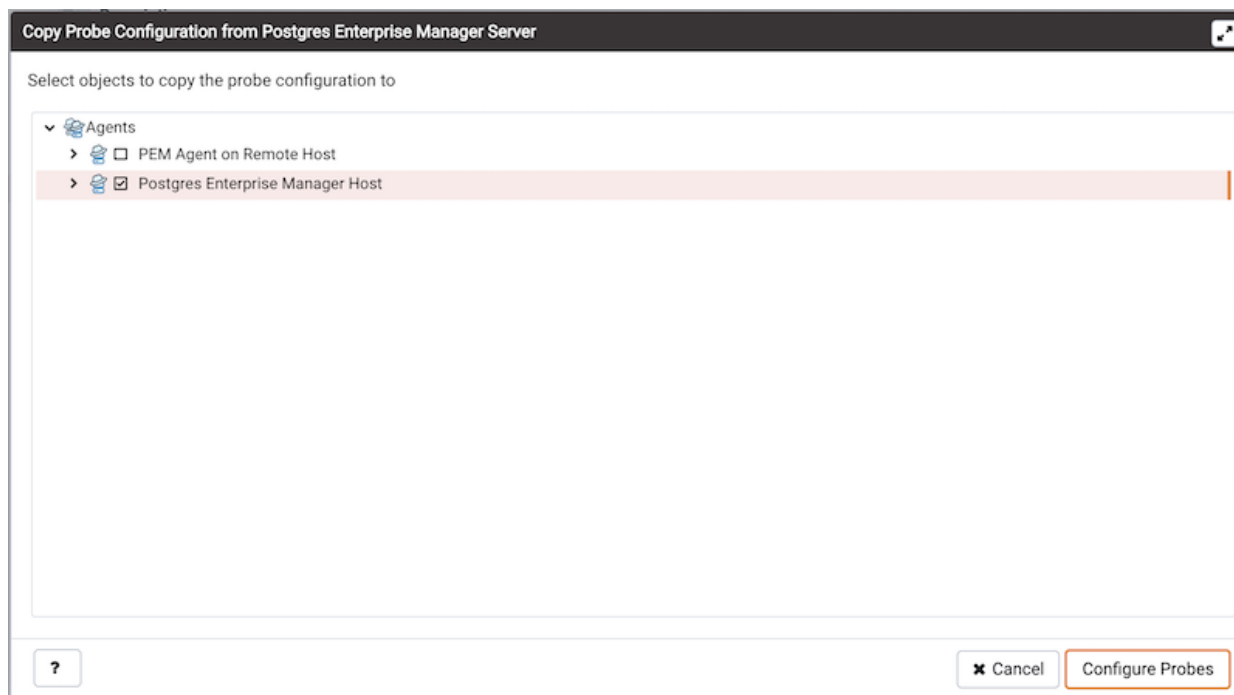
**Deleting Probes**

You may delete only user-defined probes. To delete a probe, select the probe name in the probes table, and select the **Delete** icon (located to the upper-right corner of the table). The probe history will persist for the length of time specified on the **History Retention** field in the probe definition. During the deletion the probe definition is deleted and any corresponding tables are dropped from the **pemdata** and **pemhistory** schemas.

System probes are the built-in probes provided by PEM, and are part of the PEM schema. You may only modify system probes; if you attempt to delete a system probe, you will receive an error from PEM.

**34.4.12.2 Copy Probe Configuration**

You can use the **Copy Probe Configuration...** dialog to copy probe definitions from one monitored object to one or more monitored objects of the same type. To open the **Copy Probe Configuration...** dialog, highlight the object from which you are copying probes in the PEM client tree control, and select **Manage Probes...** from the **Management** menu. When the **Manage Probes** tab opens, click on **Copy Probe** to open the **Copy Probe Configuration** dialog:



The dialog will copy the probe definitions from the object through which the **Copy Probe Configuration** dialog was opened, to the location(s) selected on the **Copy Probe Configuration** dialog tree control.

Note that if you specify a parent node in the **Copy Probe Configuration** tree control, PEM will copy the probe configurations to each object (of the same type) that resides under that node in the tree control. For example, to copy the probe definitions from one schema to all schemas that reside within a database, you only need to select the parent database of the target schemas. Please note that a red warning symbol is displayed to the left of the name of a listed target object if that object is the source of the probe that is being copied.

When you have selected the target object or objects, click the **Configure Probes** button to copy the probe definitions to the location selected on the dialog.

#### Note

At the moment the **Copy Probe** is not supported for the extension level probes.

### 34.4.12.3 Probe Configuration

A **probe** is a scheduled task that returns a set of performance metrics about a specific monitored server, database, operating system or agent. You can use the Manage Probes tab to override the default configuration and customize the behavior of each probe. To open the Manage Probes tab, select **Manage Probes...** from the **Management** menu.



Description

**Manage Custom Probes:** PEM uses probes to retrieve statistics from a monitored server, database, operating system or agent. You can view, reconfigure, delete, or create your own custom probes.

**Copy Probes:** PEM allows copying of probes from any chosen object recursively down through the object hierarchy. Click on Copy Probes to quickly copy the displayed probe configuration to a selected target.

Quick Links



Probes

| Probe name                               | Execution Frequency                 |         |         | Enabled?                            |                                     | Data Retention                      |      |
|--|-------------------------------------|---------|---------|-------------------------------------|-------------------------------------|-------------------------------------|------|
|  | Default?                            | Minutes | Seconds | Default?                            | Probe Enable?                       | Default?                            | Days |
| Background Writer Statistics             | <input checked="" type="checkbox"/> | 5       | 0       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 180  |
| Blocked Session Information              | <input checked="" type="checkbox"/> | 5       | 0       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 180  |
| Data and Log File Analysis               | <input checked="" type="checkbox"/> | 0       | 10      | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 180  |
| Database Frozen XID                      | <input checked="" type="checkbox"/> | 720     | 0       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 180  |
| Database Size                            | <input checked="" type="checkbox"/> | 30      | 0       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 180  |
| Database Statistics                      | <input checked="" type="checkbox"/> | 30      | 0       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 90   |
| Failover Manager Cluster Info            | <input checked="" type="checkbox"/> | 5       | 0       | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | 7    |
| Failover Manager Node Status             | <input checked="" type="checkbox"/> | 5       | 0       | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | 7    |
| Lock Information                         | <input checked="" type="checkbox"/> | 5       | 0       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 180  |
| Number of Prepared Transactions          | <input checked="" type="checkbox"/> | 5       | 0       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 180  |
| Number of WAL Files                      | <input checked="" type="checkbox"/> | 5       | 0       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 180  |
| Object Catalog: Database                 | <input checked="" type="checkbox"/> | 5       | 0       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 180  |
| Object Catalog: Tablespace               | <input checked="" type="checkbox"/> | 5       | 0       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 180  |
| PG HBA Conf                              | <input checked="" type="checkbox"/> | 30      | 0       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 180  |
| Server Information                       | <input checked="" type="checkbox"/> | 5       | 0       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 180  |
| Server log Configuration                 | <input checked="" type="checkbox"/> | 0       | 10      | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 180  |
| Session Information                      | <input checked="" type="checkbox"/> | 5       | 0       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 180  |
| Settings                                 | <input checked="" type="checkbox"/> | 5       | 0       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 180  |
| Streaming Replication                    | <input checked="" type="checkbox"/> | 5       | 0       | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | 180  |
| Streaming Replication Database Conflicts | <input checked="" type="checkbox"/> | 5       | 0       | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | 180  |
| Streaming Replication Lag Time           | <input checked="" type="checkbox"/> | 5       | 0       | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | 180  |
| Tablespace Size                          | <input checked="" type="checkbox"/> | 30      | 0       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 180  |
| User Information                         | <input checked="" type="checkbox"/> | 30      | 0       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 180  |
| WAL Archive Status                       | <input checked="" type="checkbox"/> | 30      | 0       | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | 180  |

The **Manage Probes** tab provides a set of **Quick Links** that you can use to create and manage probes:

- Click the **Manage Custom Probes** icon to open the **Custom Probes** tab and create or modify a custom probe.
- Click the **Copy Probes** icon to open the **Copy Probe** dialog, and copy the probe configurations from the currently selected object to one or more monitored objects.

A probe monitors a unique set of metrics for each specific object type (server, database, database object, or agent); select the name of an object in the tree control to review the probes for that object.

To modify the properties associated with a probe, highlight the name of a probe, and customize the settings that are displayed in the **Probes** table:

- Move the **Default** switch in the **Execution Frequency** columns to **No** to enable the **Minutes** and **Seconds** selectors, and specify a non-default value for the length of time between executions of the probe.
- Move the **Default** switch in the **Enabled?** column to **No** to change the state of the probe, and indicate if the probe is active or not active. If data from a probe that is **Disabled** is used in a chart, the chart will display an information icon in the upper-left corner that allows you to enable the probe by clicking the provided link.
- Move the **Default** switch in the **Data Retention** column to **No** to enable the **Day(s)** field and specify the number of days that

information gathered by the probe is stored on the PEM server.

The **Manage Probes** tab may also display information about probes that cannot be modified from the current node; if a probe cannot be modified from the current dialog, the switches for that probe are disabled. Generally, a disabled probe can be modified from a node that is higher in the hierarchy of the PEM client tree control. Select another object in the tree control to change which probes are displayed or enabled on the **Manage Probes** tab.

Contents:

34.4.12.3.1 Probes

A **probe** is a scheduled task that retrieves information about the database objects that are being monitored by the PEM agent. PEM uses the collected information to build the graphs displayed on each homepage. The **Manage Probes** tab (accessed via the **Management** menu) allows you to modify the data collection schedule and the length of time that PEM will retain information returned by a specific probe.

Unless otherwise noted, Postgres Enterprise Manager enables the following probes at the server, database, schema, extension (starting with version 8.4), or agent levels:

| Probe Name                   | Information Monitored by Probe  | Probe Configurati on Level |
|------------------------------|---|----------------------------|
| Background Writer Statistics | This probe monitors information about the background writer. The information includes: <ul style="list-style-type: none"><li>- The number of timed checkpoints</li><li>- The number of requested checkpoints</li><li>- The number of buffers written (by checkpoint)</li><li>- The number of buffers written (by background writer)</li><li>- The number of background writer cycles</li><li>- The number of background buffers written</li><li>- The number of buffers allocated</li></ul> | Server                     |
| Barman Configuration         | This probe returns information about the Barman tool global configuration.  | Agent                      |
| Barman Information           | This probe returns information about the Barman tool.   | Agent                      |
| Barman Server                | This probe returns information about the respective database server configuration monitored by Barman.  | Agent                      |
| Barman Server Status         | This probe returns information about the respective database server status monitored by Barman.   | Agent                      |
| Barman Server Backup         | This probe returns information about the backups of the respective database servers.  | Agent                      |
| Barman Server WAL Status     | This probe returns information about the Barman server WAL files.   | Agent                      |
| Blocked Session Information  | This probe returns information about the blocked sessions.  | Server                     |
| CPU Usage                    | This probe monitors CPU Usage information.  | Agent                      |
| Data and Log File Analysis   | This probe monitors information about log files. The information includes: <ul style="list-style-type: none"><li>- The name of the log file</li><li>- The directory in which the log file resides</li></ul>   | Server                     |
| Database Frozen XID          | This probe monitors the frozen XID of each database.  | Server                     |
| Database Size                | This probe monitors information about the size of the monitored databases. The information includes: <ul style="list-style-type: none"><li>- The time the information was gathered</li><li>- The database name</li><li>- The database size (in MB's)</li></ul>  | Server                     |

| Probe Name                    | Information Monitored by Probe  | Probe Configuration Level |
|-------------------------------|---|---------------------------|
| Database Statistics           | <p>This probe monitors database statistics. The information includes:</p> <ul style="list-style-type: none"> <li>- The number of backends</li> <li>- The number of transactions committed</li> <li>- The number of transactions rolled back</li> <li>- The number of blocks read</li> <li>- The number of blocks hit</li> <li>- The number of rows returned</li> <li>- The number of rows fetched</li> <li>- The number of rows inserted</li> <li>- The number of rows updated</li> <li>- The number of rows deleted</li> </ul> | Server                    |
| Disk Busy Info                | <p>This probe monitors information about disk activity.</p> <ul style="list-style-type: none"> <li>- <b>Note:</b> This probe is not supported on Mac OS X, Solaris or HP-UX</li> </ul>  | Agent                     |
| Disk Space                    | <p>This probe monitors information about disk space usage. The information includes:</p> <ul style="list-style-type: none"> <li>- The amount of disk space used</li> <li>- The amount of disk space available</li> </ul>  | Agent                     |
| EDB Audit Configuration       | This probe monitors the audit logging configuration of Postgres Plus Advanced Servers.  | Server                    |
| Failover Manager Cluster Info | This probe monitors a Failover Manager cluster, returning information about the cluster. This probe is disabled unless a cluster name and path of the Failover Manager binary is provided on the Server Properties dialog.  | Server                    |
| Failover Manager Node Status  | This probe monitors a Failover Manager cluster, returning detailed about each node within the cluster. This probe is disabled unless a cluster name and path of the Failover Manager binary is provided on the Server Properties dialog.  | Server                    |
| Function Statistics           | <p>This probe monitors a database, retrieving information about functions. The information includes:</p> <ul style="list-style-type: none"> <li>- Function names</li> <li>- Argument types</li> <li>- Return values</li> </ul>  | Database                  |
| Index Size                    | <p>This probe monitors a database, retrieving information about indexes. The information includes:</p> <ul style="list-style-type: none"> <li>- The name of the index</li> <li>- The time the data was gathered</li> <li>- The size of the index (in MB's)</li> </ul>   | Database                  |
| Index Statistics              | <p>This probe monitors index statistics. The information includes:</p> <ul style="list-style-type: none"> <li>- The number of index scans</li> <li>- The number of rows read</li> <li>- The number of rows fetched</li> <li>- The number of blocks read</li> <li>- The number of blocks hit</li> </ul>  | Database                  |
| Installed Packages            | <p>This probe monitors the packages that are currently installed. The information gathered includes:</p> <ul style="list-style-type: none"> <li>- The name of the installed package</li> <li>- The version of the installed package</li> <li>- The date and time that the probe executed</li> </ul>   | Agent                     |
| IO Analysis                   | <p>This probe monitors disk I/O information in. The information includes:</p> <ul style="list-style-type: none"> <li>- The number of blocks read</li> <li>- The number of blocks written</li> <li>- The date and time that the probe executed</li> <li>- <b>Note:</b> This probe is not supported on Mac OS X</li> </ul>  | Agent                     |
| Load Average                  | <p>This probe monitors CPU load averages. The information includes:</p> <ul style="list-style-type: none"> <li>- The 1-minute load average</li> <li>- The 5-minute load average</li> <li>- The 15-minute load average</li> <li>- <b>Note:</b> This probe is not supported on Windows</li> </ul>   | Agent                     |
| Lock Information              | <p>This probe monitors lock information. The information includes:</p> <ul style="list-style-type: none"> <li>- The database name</li> <li>- The lock type</li> <li>- The lock mode</li> <li>- The process holding the lock</li> </ul>  | Server                    |

| Probe Name                      | Information Monitored by Probe   | Probe Configuration Level |
|---------------------------------|--|---------------------------|
| Memory Usage                    | <p>This probe monitors information about system memory usage. The information includes:</p> <ul style="list-style-type: none"> <li>- Total RAM in MB</li> <li>- Free RAM in MB</li> <li>- Total swap memory in MB</li> <li>- Free swap memory in MB</li> <li>- Shared system memory in MB (It is used by tuning wizard to tune the memory parameters for the database server)</li> <li>- On non-windows system, it is <code>shmmmax</code> value and read from <code>/proc/sys/kernel/shmmmax</code></li> <li>- On windows, it is same as total memory.</li> </ul> | Agent                     |
| Network Statistics              | <p>This probe monitors network statistics. The information includes:</p> <ul style="list-style-type: none"> <li>- The interface IP address</li> <li>- The number of packets sent</li> <li>- The number of packets received</li> <li>- The number of bytes sent</li> <li>- The number of bytes received</li> <li>- The link speed (in MB/second)</li> </ul>   | Agent                     |
| Number of Prepared Transactions | This probe stores the number of prepared transactions.   | Server                    |
| Number of WAL Files             | This probe monitors the number of WAL files.   | Server                    |
| Object Catalog: Database        | <p>This probe monitors a list of databases and their properties The information includes:</p> <ul style="list-style-type: none"> <li>- The database name</li> <li>- The database encoding type</li> <li>- If the database allows user connections or system connections</li> </ul>   | Server                    |
| Object Catalog: Foreign Key     | <p>This probe monitors a list of foreign keys and their properties. The information includes:</p> <ul style="list-style-type: none"> <li>- The name of the table that contains the foreign key</li> <li>- The name of the table that the foreign key references</li> <li>- The name of the database in which the table resides</li> <li>- The name of the schema in which the table resides</li> </ul>   | Schema                    |
| Object Catalog: Function        | <p>This probe monitors a list of functions and their properties. The information includes:</p> <ul style="list-style-type: none"> <li>- The name of the function</li> <li>- The name of the schema in which the function resides</li> <li>- The name of the database in which the function resides</li> </ul>  | Schema                    |
| Object Catalog: Index           | <p>This probe monitors a list of indexes and their properties. The information includes:</p> <ul style="list-style-type: none"> <li>- The name of the index</li> <li>- The name of the table that the index is associated with</li> <li>- The name of the database in which the indexed table resides</li> </ul>   | Schema                    |
| Object Catalog: Schema          | This probe monitors a list of schemas and their associated databases and servers.  | Database                  |
| Object Catalog: Sequence        | This probe monitors a list of sequences and their properties.  | Schema                    |
| Object Catalog: Table           | <p>This probe monitors a list of table information. The information includes:</p> <ul style="list-style-type: none"> <li>- The table name</li> <li>- The name of the schema in which the table resides</li> <li>- The name of the database in which the schema resides</li> <li>- A Boolean indicator that indicates if the table has a primary key</li> </ul>   | Schema                    |
| Object Catalog: Tablespace      | This probe monitors a list of tablespaces.   | Server                    |
| Operating System Information    | This probe monitors the operating system details and boot time.  | Agent                     |
| Package Catalog                 | <p>This probe monitors the packages that are currently available for installation. The information gathered includes:</p> <ul style="list-style-type: none"> <li>- The package name</li> <li>- The package version</li> </ul>  | Agent                     |
| PG HBA Conf                     | This probe monitors authentication configuration information from the <code>pg_hba.conf</code> file.   | Server                    |
| Server Information              | This probe monitors information about servers.   | Server                    |

| Probe Name                               | Information Monitored by Probe  | Probe Configuration Level |
|--|---|---------------------------|
| Session Information                      | <p>This probe monitors session information. The information includes:</p> <ul style="list-style-type: none"> <li>- The name of the session user</li> <li>- The date and time that the session connected to the server</li> <li>- The status of the session at the time that the information was gathered (idle, waiting, etc)</li> <li>- The client address and port number</li> </ul>  | Server                    |
| Settings                                 | This probe monitors the values currently assigned to GUC variables.   | Server                    |
| SQL Protect                              | This probe monitors a server, retrieving information about SQL injection attacks.   | Server                    |
| Slony Replication                        | This probe monitors lag data for clusters replicated using Slony.   | Database                  |
| Streaming Replication                    | <p>This probe monitors a cluster that is using streaming replication, retrieving information about:</p> <ul style="list-style-type: none"> <li>- The sent Xlog location (in bytes)</li> <li>- The write Xlog location (in bytes)</li> <li>- The flush Xlog location (in bytes)</li> <li>- The replay Xlog location (in bytes)</li> <li>- The Xlog lag (in segments)</li> <li>- The Xlog lag (in pages)</li> </ul>   | Server                    |
| Streaming Replication Lag Time           | <p>This probe monitors a cluster that is using streaming replication, retrieving lag information about:</p> <ul style="list-style-type: none"> <li>- Replication lag time (in seconds)</li> <li>- Current status of replication (running/paused)</li> </ul>   | Server                    |
| Streaming Replication Database Conflicts | <p>This probe monitors a database that is using streaming replication, retrieving information about any conflicts that arise. This includes information about queries that have been canceled due to:</p> <ul style="list-style-type: none"> <li>- The # of drop tablespace conflicts</li> <li>- The # of lock timeout conflicts</li> <li>- The # of old snapshot conflicts</li> <li>- The # of pinned buffer conflicts</li> <li>- The # of deadlock conflicts</li> </ul>   | Server                    |
| Table Bloat                              | <p>This probe monitors information about the current table bloat. The information includes:</p> <ul style="list-style-type: none"> <li>- The name of the table</li> <li>- The name of the schema in which the table resides</li> <li>- The estimated number of pages</li> <li>- The estimated number of wasted pages</li> <li>- The estimated number of bytes per row</li> </ul>  | Database                  |
| Table Frozen XID                         | This probe monitors the frozen XID of each table.   | Schema                    |
| Table Size                               | <p>This probe monitors information about table size. The information includes:</p> <ul style="list-style-type: none"> <li>- Table size (in MB's)</li> <li>- Total index size (in MB's)</li> <li>- Total table size, with indexes and TOAST (in MB's)</li> </ul>   | Database                  |
| Table Statistics                         | <p>This probe monitors table statistics. The information includes:</p> <ul style="list-style-type: none"> <li>- The number of sequential scans</li> <li>- The number of sequential scan rows</li> <li>- The number of index scans</li> <li>- The number of index scan rows</li> <li>- The number of rows inserted</li> <li>- The number of rows updated</li> <li>- The number of rows deleted</li> <li>- The number of live rows</li> <li>- The number of dead rows</li> <li>- The last VACUUM</li> <li>- The last auto-vacuum</li> <li>- The last ANALYZE</li> <li>- The last auto-analyze</li> <li>- The number of pages estimated by ANALYZE</li> <li>- The number of rows estimated by ANALYZE</li> </ul> | Database                  |
| Tablespace Size                          | This probe monitors a list of tablespaces and their sizes.  | Server                    |
| User Information                         | <p>This probe monitors a list of the current users. The stored information includes:</p> <ul style="list-style-type: none"> <li>- The user name</li> <li>- The user type (superuser vs. non-superuser)</li> <li>- The server to which the user is connected</li> </ul>  | Server                    |

| Probe Name         | Information Monitored by Probe   | Probe Configurati<br>on Level |
|--------------------|--|-------------------------------|
| WAL Archive Status | This probe monitors the status of the WAL archive. The stored information includes: <ul style="list-style-type: none"><li>- The # of WAL archives done</li><li>- The # of WAL archives pending</li><li>- The last archive time</li><li>- The # of WAL archives failed</li><li>- The time of the last failure</li></ul> | Server                        |
| xDB Replication    | This probe monitors lag data for clusters replicated using xDB replication.  | Database                      |

PGD Probes

To monitor the PGD Group via [PGD dashboards](#) the following probes must be enabled. All these probes are configured at the extension level.

Note

Prior to version 8.4, all these probes are available at the server level.

PEM

File

Object

Management

Dashboards

Tools

Help

Browser

BARMAN Servers

BART Servers

PEM Agents

PEM Server Directory (2)

Postgres Enterprise Manager Server

unwrap

Databases (2)

bdrdb

Casts

Catalogs

Event Triggers

Extensions (10)

bdr

bdr-enterprise

btree\_gist

pagesinspect

pg\_freespacemap

pg\_stat\_statements

pg\_visibility

pglogical

pgstattuple

plpgsql

Foreign Data Wrappers

Languages

Publications

Schemas

Subscriptions

postgres

Login/Group Roles (18)

Tablespaces

Dashboard

Properties

SQL

Statistics

Dependencies

Dependents

Monitoring

Manage Probes

Probes

enterpriseDB

Description

Manage Custom Probes: PEM uses probes to retrieve statistics from a monitored server, database, operating system or agent. You can view, reconfigure, delete, or create your own custom probes.

Copy Probes: PEM allows copying of probes from any chosen object recursively down through the object hierarchy. Click on Copy Probes to quickly copy the displayed probe configuration to a selected target.

Quick Links

Manage Custom Probes

Copy Probes

Help

Probes

| Probe name                          | Execution Frequency |         | Enabled? |                | Data Retention |                   |
|-------------------------------------|---------------------|---------|----------|----------------|----------------|-------------------|
|                                     | Default?            | Minutes | Seconds  | Default?       | Probe Enable?  | Days              |
| PGD Conflict History Summary        | <div>Yes</div>      | 1       | 0        | <div>Yes</div> | <div>No</div>  | <div>Yes</div> 7  |
| PGD Global Locks                    | <div>Yes</div>      | 1       | 0        | <div>Yes</div> | <div>No</div>  | <div>Yes</div> 30 |
| PGD Group Camo Details              | <div>Yes</div>      | 1       | 0        | <div>Yes</div> | <div>No</div>  | <div>Yes</div> 30 |
| PGD Group Raft Details              | <div>Yes</div>      | 1       | 0        | <div>Yes</div> | <div>No</div>  | <div>Yes</div> 30 |
| PGD Group Replication Slots Details | <div>Yes</div>      | 1       | 0        | <div>Yes</div> | <div>No</div>  | <div>Yes</div> 30 |
| PGD Group Subscription Summary      | <div>Yes</div>      | 0       | 10       | <div>Yes</div> | <div>No</div>  | <div>Yes</div> 30 |
| PGD Group Versions Details          | <div>Yes</div>      | 1440    | 0        | <div>Yes</div> | <div>No</div>  | <div>Yes</div> 30 |
| PGD Monitor Group Raft Status       | <div>Yes</div>      | 1       | 0        | <div>Yes</div> | <div>No</div>  | <div>Yes</div> 30 |
| PGD Monitor Group Versions          | <div>Yes</div>      | 1440    | 0        | <div>Yes</div> | <div>No</div>  | <div>Yes</div> 30 |
| PGD Node Replication Rates          | <div>Yes</div>      | 0       | 10       | <div>Yes</div> | <div>No</div>  | <div>Yes</div> 30 |
| PGD Node Slots                      | <div>Yes</div>      | 0       | 10       | <div>Yes</div> | <div>No</div>  | <div>Yes</div> 30 |
| PGD Node Summary                    | <div>Yes</div>      | 1       | 0        | <div>Yes</div> | <div>No</div>  | <div>Yes</div> 30 |
| PGD Stat Relation                   | <div>Yes</div>      | 1       | 0        | <div>Yes</div> | <div>No</div>  | <div>Yes</div> 30 |
| PGD Stat Subscription               | <div>Yes</div>      | 1       | 0        | <div>Yes</div> | <div>No</div>  | <div>Yes</div> 30 |
| PGD Worker Errors                   | <div>Yes</div>      | 0       | 15       | <div>Yes</div> | <div>No</div>  | <div>Yes</div> 30 |
| PGD Workers                         | <div>Yes</div>      | 1       | 0        | <div>Yes</div> | <div>No</div>  | <div>Yes</div> 30 |

The user with `pgd_superuser` will be able to view information from all the following probes.

All the following probes works with `PGD Enterprise Edition`.

| Probe Name                   | Information Monitored by Probe   | <code>pgd_monitor</code> role required | Works with PGD SE? |
|------------------------------|--|--|--------------------|
| PGD Conflict History Summary | This probe returns information about row conflicts per conflict type. The stored information includes: <ul style="list-style-type: none"><li>- The local time of the conflict</li><li>- The type of the conflict</li></ul> | Yes                                    | Yes                |

| Probe Name                         | Information Monitored by Probe   | pgd_monitor role required | Works with PGD SE? |
|------------------------------------|--|---------------------------|--------------------|
| PGD Global Locks                   | <p>This probe returns information about global locks in a PGD Group. The stored information includes:</p> <ul style="list-style-type: none"> <li>- The name of the node where the global lock has originated</li> <li>- The PID of the process holding the lock</li> <li>- The type of lock (DDL or DML)</li> <li>- The name of the locked relation(for DML Locks) or Keys(for advisory locks)</li> <li>- The internal state of the lock acquisition process</li> <li>- The list of backends waiting for the same global lock</li> <li>- The time when the global lock acquire was initiated by origin node</li> <li>- The time when the local node started trying to acquire the local lock</li> <li>- The time acquire_stage last changed</li> </ul>   | Yes                       | Yes                |
| PGD Group Camo Details             | <p>This probe returns information about Camo's in PGD Group. The stored information includes:</p> <ul style="list-style-type: none"> <li>- The name of the node</li> <li>- The node name for whom this node is partner</li> <li>- The node name for whom this node is origin</li> <li>- The connection status</li> <li>- The readiness status</li> <li>- The number of pending or unresolved camo transactions</li> <li>- The lsn of last applied wal log</li> <li>- The lsn of last received wal log</li> </ul>   | No                        | No                 |
| PGD Group Replication Slot Details | <p>This probe returns information about replication slots in PGD Group. The stored information includes:</p> <ul style="list-style-type: none"> <li>- The name of the PGD Group</li> <li>- The name of the origin node</li> <li>- The name of the target node</li> <li>- The slot name on the origin node used by this subscription</li> <li>- The active status</li> <li>- The state of the replication (catchup,streaming,disconnected,...)</li> <li>- The approximate lag time for reported write</li> <li>- The approximate lag time for reported flush</li> <li>- The approximate lag time for reported replay</li> <li>- The bytes difference between sent_lsn and current WAL write position</li> <li>- The bytes difference between write_lsn and current WAL write position</li> <li>- The bytes difference between flush_lsn and current WAL write position</li> <li>- The bytes difference between replay_lsn and current WAL write position</li> </ul> | No                        | Yes                |
| PGD Group Subscription Summary     | <p>This probe returns information about the summary of Subscriptions in the PGD Group. The stored information includes:</p> <ul style="list-style-type: none"> <li>- The name of the origin of the subscription</li> <li>- The name of the target of the subscription</li> <li>- The timestamp of the last replayed transaction</li> <li>- The lag between now and time of last replayed transaction</li> </ul>  | No                        | Yes                |
| PGD Monitor Group Raft             | This probe returns the status and message of a cluster-wide raft check.  | Yes                       | Yes                |
| PGD Group Raft Details             | <p>This probe returns the information about raft consensus status from all the nodes in PGD Group. The stored information includes:</p> <ul style="list-style-type: none"> <li>- The name of the node</li> <li>- The raft worker state on the node</li> <li>- The node id of the RAFT_LEADER</li> <li>- The raft election internal id</li> <li>- The raft snapshot internal id</li> </ul>  | No                        | Yes                |
| PGD Monitor Group Versions         | This probe returns the status and message of cluster-wide version check.   | Yes                       | Yes                |
| PGD Group Versions Details         | <p>This probe returns the information about version details of the installed postgres, pglogical, pgd, and PGD edition for each node in the PGD Group. The stored information includes:</p> <ul style="list-style-type: none"> <li>- The name of the node</li> <li>- The installed postgres version on the node</li> <li>- The installed pglogical version on the node</li> <li>- The version of the PGD on the node</li> <li>- The PGD edition (Standard/Enterprise)</li> </ul>   | No                        | Yes                |



| Probe Name                 | Information Monitored by Probe  | pgd_monitor role required | Works with PGD SE? |
|----------------------------|---|---------------------------|--------------------|
| PGD Node Replication Rates | <p>This probe returns information about outgoing replication activity from a given node. The stored information includes:</p> <ul style="list-style-type: none"> <li>- The name of the target peer node</li> <li>- The latest sent position</li> <li>- The latest position reported as replayed</li> <li>- The approximate lag time for reported replay</li> <li>- The bytes difference between replay_lsn and current WAL write position on origin</li> <li>- The human readable bytes difference between replay_lsn and current WAL write position</li> <li>- Approximate time required for the peer node to catchup to all the changes that are yet to be applied</li> </ul>   | Yes                       | No                 |
| PGD Node Slots             | <p>This probe returns information about the mapping of local PGD nodes to replication slots, their status, and replication progress. The stored information includes:</p> <ul style="list-style-type: none"> <li>- The name of the slot</li> <li>- The name of the target node</li> <li>- The name of the PGD Group</li> <li>- The database name on the target node</li> <li>- The PID of the process attached to the slot</li> <li>- The catalog XID needed by the slot</li> <li>- The ip address of the client connection</li> <li>- The latest sent position</li> <li>- The latest position reported as replayed</li> <li>- The approximate lag time for reported replay</li> <li>- The Bytes difference between replay_lsn and current WAL write position</li> <li>- The Human-readable bytes difference between replay_lsn and current WAL write position</li> </ul> | Yes                       | Yes                |
| PGD Node Summary           | <p>This probe returns information about all the nodes in the PGD Group. The stored information includes:</p> <ul style="list-style-type: none"> <li>- The name of the Node</li> <li>- The name of the PGD group the node is part of</li> <li>- The consistent state of the node in human readable form</li> <li>- The state which the node is trying to reach (during join or promotion)</li> <li>- The name of subscribed repsets</li> </ul>   | Yes                       | Yes                |
| PGD Workers                | <p>This probe returns information about workers in PGD node. The stored information includes:</p> <ul style="list-style-type: none"> <li>- The PID of the worker process</li> <li>- The worker query start timestamp</li> <li>- The worker state change timestamp</li> <li>- The worker wait event type</li> <li>- The worker wait event</li> <li>- The worker state</li> <li>- The worker role name</li> <li>- The worker commit timestamp</li> <li>- The worker local timestamp</li> <li>- The name of the origin node</li> <li>- The receive LSN</li> <li>- The receive commit LSN</li> <li>- The last exact replay LSN</li> <li>- The last exact flush LSN</li> <li>- The last exact replay timestamp</li> <li>- The worker query</li> </ul>  | Yes                       | Yes                |

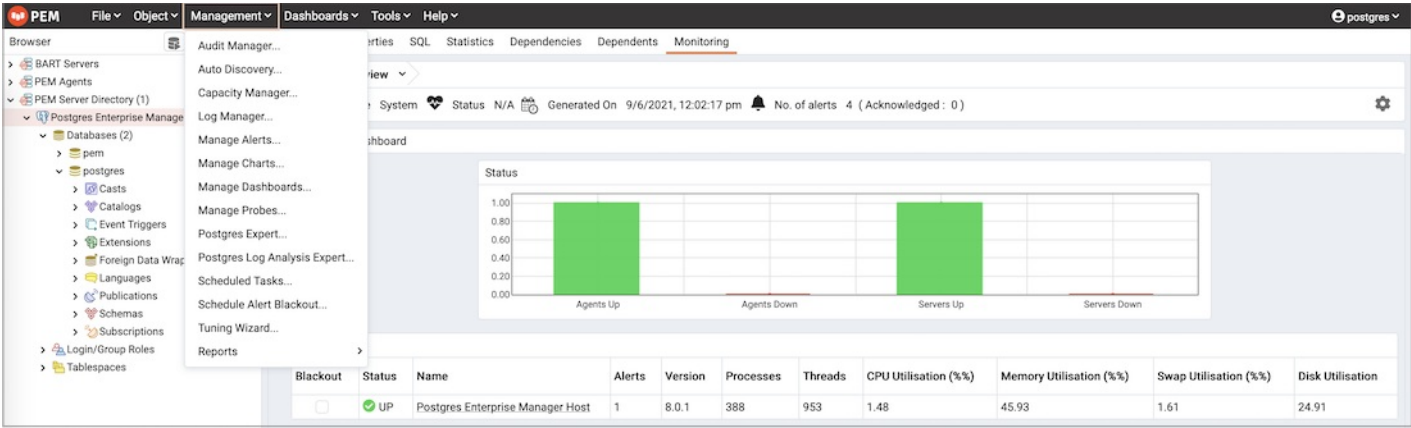


| Probe Name        | Information Monitored by Probe  | pgd_monitor role required | Works with PGD SE? |
|-------------------|---|---------------------------|--------------------|
| PGD Worker Errors | <div>This probe returns information about the worker errors in PGD node. The stored information includes:<ul style="list-style-type: none"><li>- The process id of the worker causing the error</li><li>- The name of the PGD Group the node is part of</li><li>- The name of the origin node</li><li>- The name of the source node</li><li>- The name of the target node</li><li>- The name of the subscription</li><li>- The internal identifier of the role of this worker</li><li>- The name of the role of this worker</li><li>- The date and time of the error</li><li>- The age of the error</li><li>- The description of the error</li><li>- The context in which the error happened</li><li>- The remote relation id</li><li>- The subscription writer id</li><li>- The subscription writer name</li></ul></div> | Yes                       | Yes                |

34.4.13 Schedule Alert Blackout

You can use the **Schedule Alert Blackout** option on the **Management** menu to schedule an alert blackout for your Postgres servers and PEM Agents during maintenance. Alerts will not be raised during a defined blackout period.

To schedule an alert blackout, click on the **Management** menu and select **Schedule Alert Blackout**.



When the **Schedule Alert Blackout** dialog opens, use the tabs on the dialog to define the blackout period for servers and agents. Open the **Server** tab and click the Add icon (+) at the top right corner to add new row.

Schedule Alert Blackout

ServerAgent

Schedule Blackouts

☐

Start time

2020-09-18 01:20 -07:00

2020-09-30 23:27 -07:00

☒2020-09-28 23:27 -07:00

Duration

1 hour

6 hours

4 hours

Servers

Postgres Enterprise Manager Server

Postgres Enterprise Manager Server

Postgres Enterprise Manager Server

×

×

×

?

×

Cancel

↺

Reset

💾

Save

Use the fields on the **Server** tab to provide information about a Server to blackout the alerts:

- Use the **Start time** field to provide the date and time to start the alert blackout.
- Use the **Duration** field to provide the interval for which you want to blackout the alerts.
- Use the **Servers** field to provide the server name for which you want to blackout the alerts. You can also select multiple servers to blackout the alerts at same time.

Once all the details are provided, you can save the details by clicking on **Save** button on the right bottom corner of the dialog. Once saved, it cannot be edited. The alerts will not be displayed on the Alerts dashboard for the scheduled interval of that particular server.

You can also schedule a blackout period for PEM Agents via the **Agent** tab on the dialog. Open the **Agent** tab and click the Add icon (+) at the top right corner to add new row.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

564

Schedule Alert Blackout

Server

Agent

Schedule Blackouts

Start time

Duration

Agents

2020-09-18 03:33 -07:00

1 hour

Postgres Enterprise Manager Host

2020-09-25 03:33 -07:00

4 hours

Agent2

2020-09-30 03:33 -07:00

23 hours

Postgres Enterprise Manager Host

2020-09-21 23:23 -07:00

6 hours

Agent2

?

Cancel

Reset

Save

Use the fields on the **Agent** tab to provide the information about an Agent to blackout the alerts:

- Use the **Start time** field to provide the date and time to start the alert blackout.
- Use the **Duration** field to provide the interval for which you want to blackout the alerts.
- Use the **Agents** field to provide the Agent name for which you want to blackout the alerts. All server level alerts, for the servers bind to that particular agent will blackout.

Once all the details are provided, you can save the details by clicking on **Save** button on the right bottom corner of the dialog. Once saved, it cannot be edited. The alerts will not be displayed on the Alert dashboard for the scheduled interval for that **PEM Agent** .

You can use **Clone** button from the top right corner of dialog, to clone the scheduling of alert blackout. Select the servers or agents you want to clone and then click on **Clone** button to create the cloned copy of all the selected servers or agents. You can edit newly created schedules as needed, and then click **Save** .

You can use **Delete** button from the top right corner of dialog to remove a scheduled alert blackout. Select the servers or agents and then click on highlighted **Delete** button in the right top corner to remove the scheduled alerts associated with that server or agent.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

565

Schedule Alert Blackout

ServerAgent

Schedule Blackouts

| <input type="checkbox"/>            | Start time              | Duration | Servers                                       |
|-------------------------------------|-------------------------|----------|---|
| <input type="checkbox"/>            | 2020-09-18 01:20 -07:00 | 1 hour   | <div>Postgres Enterprise Manager Server</div> |
| <input checked="" type="checkbox"/> | 2020-09-30 23:27 -07:00 | 6 hours  | <div>Postgres Enterprise Manager Server</div> |
| <input checked="" type="checkbox"/> | 2020-09-28 23:27 -07:00 | 4 hours  | <div>Postgres Enterprise Manager Server</div> |

?

Cancel

Reset

Save

Select a server for which you wish to delete the scheduled alert backout and then click on the `Delete` button. The server will ask for confirmation before deleting that row.

Schedule Alert Blackout

Alert Blackout

ServerAgent

All the selected rows will be deleted, Are you wish to continue?

Cancel

OK

Schedule Blackouts

|                                     |                         |         |   |
|-------------------------------------|-------------------------|---------|---|
| <input type="checkbox"/>            | 2020-09-18 01:20 -07:00 | 1 hour  | <div>Postgres Enterprise Manager Server</div> |
| <input type="checkbox"/>            | 2020-09-30 23:27 -07:00 | 6 hours | <div>Postgres Enterprise Manager Server</div> |
| <input checked="" type="checkbox"/> | 2020-09-28 23:27 -07:00 | 4 hours | <div>Postgres Enterprise Manager Server</div> |

?

Cancel

Reset

Save

You can use the **Reset** button to reset the details on the Alert Blackout dialog to the default settings. Please note that all saved blackouts will remain unaffected after resetting the current dialog values.

34.4.14 Scheduled System Jobs

PEM defines system jobs to take care of cleanup activities at scheduled intervals. All of the system jobs are enabled by default, and are scheduled to deploy on a regular interval. You can query the `pem.job` table in the `pem` database to review a list of the system jobs. The current schedule for system jobs is stored in the `pem.schedule` table in the `pem` database.

The system job names, their descriptions, and default deployment intervals are listed in the table below:

| Job Name                     | Description   | Intervals to Run the Job   |
|------------------------------|---|--|
| Database cleanup             | This job runs periodically to purge old data from the database.   | Every four hours, daily starting from 01:00 AM   |
| Audit log table cleanup      | This job runs periodically to purge old data from the audit log table.  | Every twelve hours, daily starting from 12:00 AM   |
| Server log table cleanup     | This job runs periodically to purge old data from the server log table.   | Every twelve hours, daily starting from 01:00 AM   |
| Probe log table cleanup      | This job runs periodically to purge old data from the probe log table.  | Every twelve hours, daily starting from 02:00 AM   |
| SMTP spool table cleanup     | This job runs periodically to purge old data from the smtp spool table.   | Every twelve hours, starting from 03:00 AM   |
| SNMP spool table cleanup     | This job runs periodically to purge old data from the snmp spool table.   | Every twelve hours, daily starting from 04:00 AM   |
| Alert history table cleanup  | This job runs periodically to purge old data from the alert history table.  | Every twelve hours, daily starting from 05:00 AM   |
| Job log table cleanup        | This job runs periodically to purge old data from the job log table.  | Every twelve hours, daily starting from 06:00 AM   |
| Job purge the deleted charts | This job runs periodically to purge the deleted charts.   | Every twenty-four hours, daily starting from 02:00 AM  |
| Purge deleted custom probes  | This job runs periodically to purge deleted custom probes and its data.   | Every twenty-four hours, daily starting from 01:00 AM  |
| Check CA certificate expiry  | This job check the expiry of CA certificate.  | <div>This job runs before 20 days of expiry of the certificate</div> <div>Note: PEM self signed CA Certificate is valid till 3650 days from the day of fresh installation.</div> |
| Delete obsolete backups      | This job runs periodically to delete the BART obsolete backups on database server where BART is configured only when enabled. | Every twenty-four hours, daily starting from 02:00 AM once enabled   |

34.4.15 Scheduled Task Tab

You can schedule the execution of user-defined tasks on registered servers for a time that is most convenient, and least intrusive to your users. Tasks may be one-off, or recurring and are comprised of one or more steps, which may be a SQL script, a batch/shell script, or an internal function in the PEM agent. You can view pending tasks on the *Scheduled Tasks tab*.

To open the **Scheduled Tasks** tab, select either a PEM Agent or a managed server in the tree control of the PEM client and select **Scheduled Tasks...** from the **Management** menu.

Properties SQL Statistics Dependencies Dependents Monitoring Manage Probes **Scheduled Tasks**

**Description**

**Scheduled Tasks:** Use the Scheduled Tasks tab to review and modify the scheduled tasks for a specific database or server, or for the servers monitored by an agent. Click the Execution icon to the left of a task name to review each step and the execution results. Click the Edit button to the left of a task name to review detailed information about the status and schedule of the task.

For more information, please see the online [help](#).

**Legend**

Running: Never ran:   
 Successfully finished: No steps to execute:   
 Failed: Aborted:

**Tasks**

Manage Tasks Show system tasks? ☐ No

|  | Logs | Status | Enabled?                            | Name                                  | Agent                            | Owner    |
|--|------|--------|-------------------------------------|---------------------------------------|----------------------------------|----------|
|  |      |        | <input checked="" type="checkbox"/> | PEM Log Manager Log Import - Server 1 | Postgres Enterprise Manager Host | postgres |
|  |      |        | <input checked="" type="checkbox"/> | Server log configuration request      | Postgres Enterprise Manager Host | postgres |
|  |      |        | <input checked="" type="checkbox"/> | Server log configuration request      | Postgres Enterprise Manager Host | postgres |

The tab features a legend, displaying the icons that identify the status of each task.

The **Manage Tasks** table displays a list of tasks. Set **Show system tasks?** to **Yes** to display system tasks; if it is set to **No**, only user-defined tasks are displayed. System tasks are displayed with a grey background, and may not be modified.

Use the **Refresh** icon to update the list of tasks displayed in the table. The table displays general information about each task.

- The **Execution** drop-down provides access to detailed information about each step in the task.
- The **Status** field lists the status of the current task.
- The **Enabled?** switch displays **Yes** if the task is enabled; **No** if the task is disabled.
- The **Name** field displays the name of the task.
- The **Agent** or **Server** field displays the name of the agent responsible for executing the task or the server on which the task will execute.

Highlight the name of a user-defined task and click the **Edit** icon (to the left of a task name) to access detailed information about the selected task.

**Tasks**

Manage Tasks Show system tasks? ☐ No

|  | Logs | Status | Enabled?                            | Name              | Agent                            | Owner    |
|--|------|--------|-------------------------------------|-------------------|----------------------------------|----------|
|  |      |        | <input checked="" type="checkbox"/> | Job_backup_pem_db | Postgres Enterprise Manager Host | postgres |

**General**

Status: Never ran

Enabled?: ☒ Yes

Owner: postgres

Name: Job\_backup\_pem\_db

Agent: Postgres Enterprise Manager Host

Description: Job for taking backup of pem database

Last run:

Next run:

Created: 2020-04-28 15:38:07

The **General** tab displays information about the scheduled task:

- The **Status** field lists the status of the current task.
- The **Enabled?** switch displays **Yes** if the task is enabled; **No** if the task is disabled.

- The **Name** field displays the name of the task.
- The **Agent** or **Server** field displays the name of the agent responsible for executing the task or the server on which the task will be performed.
- The **Description** field displays a description of the task.
- The **Last run** field displays the date and time of the last execution of the task.
- The **Next run** field displays the date and time of the next scheduled execution of the task.
- The **Created** field displays the date and time that the task was defined.

Highlight the name of a user-defined task and open the **Steps** arrow to review a list of the steps within the task.

Tasks

Manage Tasks

Show system tasks? ☐ No

|  | Logs | Status | Enabled? | Name                          | Agent                                 | Owner                            |
|--|------|--------|----------|-------------------------------|---------------------------------------|----------------------------------|
|  |      |        |          | <input type="checkbox"/> True | Job_backup_pem_db                     | Postgres Enterprise Manager Host |
|  |      |        |          | <input type="checkbox"/> True | PEM Log Manager Log Import - Server 1 | Postgres Enterprise Manager Host |
|  |      |        |          | <input type="checkbox"/> True | Server log configuration request      | Postgres Enterprise Manager Host |
|  |      |        |          | <input type="checkbox"/> True | Server log configuration request      | Postgres Enterprise Manager Host |

Steps

Execution date

Description

2020-04-27

Server ID: 1, agent ID: 1

| Step                   | Type     | Status | Result | Start / Next run    | Duration        | Output | Log details |
|------------------------|----------|--------|--------|---------------------|-----------------|--------|-------------|
| Modify postgresql.conf | Internal |        | 0      | 2020-04-27 14:32:49 | 00:00:00.223211 |        |             |
| Server Restart         | Internal |        | 0      | 2020-04-27 14:32:49 | 00:00:13.287379 |        |             |

The list of steps displays general information about each step in the task:

- The **Execution date** field displays the date on which the step will execute. Step history is grouped by execution date; use the arrow to the left of an execution date to expand the node and review the task logs for that date.
- The **Description** field displays a description of the step.

Use the arrow to the left of an execution date (in the **Steps** column) to view detailed information about the step:

- The **Step** field displays a description of the step.
- The **Type** field displays the task type.
- The **Status** field lists the status of the current task.
- If applicable, the **Result** field displays code generated during the execution of the step.
- The **Start/Next run** field displays the date and time at which the task executed or will execute again.
- The **Duration** field displays the length of time that the task required for execution.
- The **Output** field displays the result set returned by the execution of the task. By default, it displays the first 250 characters. You can also change the display characters by changing the **Schedule Tasks** options in the **Preference** dialog.
- The **Log details** field allows you to open the log in the new browser window and also download the complete log.

To delete a user-defined task, highlight the name of the task, and click the **Delete** icon located to the left of a task's name. The task will be marked for deletion, and removed when you click the **Save** icon (located in the upper-right corner of the **Manage Tasks** table).

**Note:** Tasks with no **Next run** date will automatically be removed from the PEM server when the last run date is more than `probe_log_retention_time` days ago.

Please note that if any of the scheduled tasks for backup, restore, validate host, validate server or delete obsolete backup for any of the BART Server gets deleted, it will not display under the **BART Tool Activities** graph of BART Server's dashboard. However, it gets listed under the **Initiated Server Backups** list.

34.4.16      Creating a PEM Scheduled Job

You can create a PEM scheduled job to perform a set of custom-defined steps in the specified sequence. These steps may contain SQL code or a batch/shell script that you may run on a server that is bound with the agent. You can schedule these jobs to suit your business requirements. For example, you can create a job for taking a backup of a particular database server and schedule it to run on a specific date and time of every month.

To create or manage a PEM scheduled job, use the PEM tree control to browse to the PEM agent for which you want to create the job. The tree control will display a Jobs node, under which currently defined jobs are displayed. To add a new job, right click on the Jobs node, and select Create Job... from the context menu.

When the Create Agent Job dialog opens, use the tabs on the Create - Agent Job dialog to define the steps and schedule that make up a PEM scheduled job.

**Create - Agent Job**

General Steps Schedules Notifications SQL

Name: Job\_backup\_pem

Enabled? ☒ Yes

Comment: Job for taking backup of pem database

Buttons: [i] [?] [Cancel] [Reset] [Save]



Use the fields on the **General** tab to provide general information about a job:

- Provide a name for the job in the **Name** field.
- Move the **Enabled** switch to the **Yes** position to enable a job, or **No** to disable a job.
- Use the **Comment** field to store notes about the job.



Create - Agent Job

GeneralStepsSchedulesNotificationsSQL

|   | Name         | Enabled?        | Kind           | On error |
|---|--------------|-----------------|----------------|----------|
|   | Step1_backup | <div>True</div> | <div>SQL</div> | Success  |

GeneralCode

Name

Step1\_backup

Enabled?

Yes

Kind

SQL

On error

Success



Server

Postgres Enterprise Manager Server (192.168.1.19:5432)

Database

pem

Comment



Cancel

Reset

Save

Use the **Steps** tab to define and manage the steps that the job will perform. Click the Add icon (+) to add a new step; then click the compose icon (located at the left side of the header) to open the step definition dialog:

**Create - Agent Job**

General Steps Schedules Notifications SQL

Name: Job\_backup\_pem\_db

Enabled? ☒ Yes

Comment: Job for taking backup of pem database

Buttons: [i] [?] [Cancel] [Reset] [Save]

Use fields on the step definition dialog to define the step:

- Provide a name for the step in the **Name** field; please note that steps will be performed in alphanumeric order by name.
- Use the **Enabled** switch to include the step when executing the job ( **True** ) or to disable the step ( **False** ).
- Use the **Kind** switch to indicate if the job step invokes SQL code ( **SQL** ) or a batch script ( **Batch** ).

- If you select **SQL** , use the **Code** tab to provide SQL code for the step.
- If you select **Batch** , use the **Code** tab to provide the batch script that will be executed during the step.

Use the **On error** drop-down to specify the behavior of pgAgent if it encounters an error while executing the step. Select from:

- **Fail** - Stop the job if you encounter an error while processing this step.
- **Success** - Mark the step as completing successfully, and continue.
- **Ignore** - Ignore the error, and continue.
- If you have selected SQL as your input for **Kind** switch, provide the following additional information:
  - Use the **Server** field to specify the server that is bound with the agent for which you are creating the PEM scheduled job.
  - Use the **Database** field to specify the database that is associated with the server that you have selected.
- Use the **Comment** field to provide a comment about the step.

Create - Agent Job

GeneralStepsSchedulesNotificationsSQL

|                                   | Name         | Enabled?        | Kind           | On error |
|-----------------------------------|--------------|-----------------|----------------|----------|
| <div><div></div><div></div></div> | step1_backup | <div>True</div> | <div>SQL</div> | Success  |

GeneralCode

Name

step1\_backup

Enabled?

Yes

Kind

SQL

On error

Success

Server

Postgres Enterprise Manager Server (192.168.1.19:5432)

Database

pem

Comment

i?

CancelResetSave

Use the context-sensitive field on the step definition dialog's `Code` tab to provide the SQL code or batch script that will be executed during the step:

- If the step invokes SQL code, provide one or more SQL statements in the `SQL query` field.
- If the step invokes a batch script, provide the script in the `Code` field. If you are running on a Windows server, standard batch file syntax must be used. When running on a Linux server, any shell script may be used, provided that a suitable interpreter is specified on the first line (e.g. `#!/bin/sh`). Along with the defined inline code, you can also provide the path of any batch script, shell script, or SQL file on the filesystem.

To invoke a script on a Linux system, you must modify the entry for `batch_script_user` parameter of `agent.cfg` file and specify the user that should be used to run the script. You can either specify a non-root user or root for this parameter. If you do not specify a user, or the specified user does not exist, then the script will not be executed. Restart the agent after modifying the file. If `pemagent` is being run by a non-root user then the value of `batch_script_user` will be ignored and the script will be executed by the same non-root user that is being used for running the `pemagent`.

To invoke a script on a Windows system, set the registry entry for `AllowBatchJobSteps` as true and restart the PEM agent. PEM registry entries are located in `HKEY_LOCAL_MACHINE\Software\EnterpriseDB\PEM\agent`.

After providing all the information required by the step, click the `Save` button to save and close the step definition dialog.

Click the add icon (+) to add each additional step, or select the `Schedules` tab to define the job schedule.


Note

When you create the Job, the time fields under `Schedules` tab takes the timezone of the client machine. While saving the jobs on the PEM Server, the timezone gets converted according to PEM Server's timezone irrespective of PEM Agent's location.

Click the Add icon (+) to add a schedule for the job; then click the compose icon (located at the left side of the header) to open the schedule definition dialog:

Create - Agent Job

GeneralStepsSchedulesNotificationsSQL

| Name   | Enabled?        | Start                   | End                     |
|--|-----------------|-------------------------|-------------------------|
|  Backup_pem_db_schedule | <div>True</div> | 2020-04-01 23:30 +05:30 | 2020-04-02 23:30 +05:30 |

GeneralRepeatExceptions

Name

Backup\_pem\_db\_schedule

Enabled?

Yes

Start

2020-04-01 23:30 +05:30

End

2020-04-02 23:30 +05:30

Comment

Cancel

Reset

Save

Use the fields on the Schedules definition tab to specify the days and times at which the job will execute.

- Provide a name for the schedule in the **Name** field.
- Use the **Enabled** switch to indicate that pgAgent should use the schedule (**Yes**) or to disable the schedule (**No**).
- Use the calendar selector in the **Start** field to specify the starting date and time for the schedule.
- Use the calendar selector in the **End** field to specify the ending date and time for the schedule.
- Use the **Comment** field to provide a comment about the schedule.

Select the **Repeat** tab to define the days on which the schedule will execute.

**Create - Agent Job**

General Schedules Notifications SQL

| Name                   | Enabled? | Start                   | End                     |
|------------------------|----------|-------------------------|-------------------------|
| Backup_pem_db_schedule | True     | 2020-04-01 23:30 +05:30 | 2020-04-02 23:30 +05:30 |

General Repeat Exceptions

Schedules are specified using a **cron-style** format.  
 For each selected time or date element, the schedule will execute.  
 e.g. To execute at 5 minutes past every hour, simply select '05' in the Minutes list box.  
 Values from more than one field may be specified in order to further control the schedule.  
 e.g. To execute at 12:05 and 14:05 every Monday and Thursday, you would click minute 05, hours 12 and 14, and weekdays Monday and Thursday.  
 For additional flexibility, the Month Days check list includes an extra Last Day option. This matches the last day of the month, whether

**Days**

Week Days

Month Days

Months

**Times**

Hours

Minutes

Buttons: [i] [?] [Cancel] [Reset] [Save]

Use the fields on the **Repeat** tab to specify the details about the schedule in a cron-style format. The job will execute on each date or time element selected on the **Repeat** tab.

Click within a field to open a list of valid values for that field; click on a specific value to add that value to the list of selected values for the field. To clear the values from a field, click the X located at the right-side of the field.

Use the fields within the **Days** box to specify the days on which the job will execute:

- Use the **Week Days** field to select the days on which the job will execute.
- Use the **Month Days** field to select the numeric days on which the job will execute. Specify the **Last Day** to indicate that the job should be performed on the last day of the month, regardless of the date.
- Use the **Months** field to select the months in which the job will execute.

Use the fields within the **Times** box to specify the times at which the job will execute:

- Use the **Hours** field to select the hour at which the job will execute.
- Use the **Minutes** field to select the minute at which the job will execute.

Select the **Exceptions** tab to specify any days on which the schedule will **not** execute.

Create - Agent Job


General

Steps

Schedules

Notifications


SQL



|   | Name                   | Enabled?        | Start                   | End                     |
|---|------------------------|-----------------|-------------------------|-------------------------|
|  | Backup_pem_db_schedule | <div>True</div> | 2020-04-01 23:30 +05:30 | 2020-04-02 23:30 +05:30 |

General

Repeat

Exceptions

|   | Date       | Time  |
|---|------------|-------|
|  | 2020-04-01 | 00:00 |



Cancel

Reset

Save

Use the fields on the **Exceptions** tab to specify days on which you wish the job to not execute; for example, you may wish for jobs to not execute on national holidays.

Click the Add icon (+) to add a row to the exception table, then:

- Click within the **Date** column to open a calendar selector, and select a date on which the job will not execute. Specify **<Any>** in the **Date** column to indicate that the job should not execute on any day at the time selected.
- Click within the **Time** column to open a time selector, and specify a time on which the job will not execute. Specify **<Any>** in the **Time** column to indicate that the job should not execute at any time on the day selected.

Select the **Notifications** tab to configure the email notification settings on job level:

The screenshot shows the 'Create - Agent Job' dialog box with the 'Notifications' tab selected. The dialog has a title bar 'Create - Agent Job' and a close button. Below the title bar are tabs: 'General', 'Steps', 'Schedules', 'Notifications' (active), and 'SQL'. The 'Notifications' tab contains two main sections: 'Send the notifications' and 'Email group'. The 'Send the notifications' section has a dropdown menu set to 'ALWAYS'. Below this dropdown is explanatory text: 'Determines when to send a notification for the job:'. It lists three options: 'ON FAILURE : Send a notification on the failure/interruption of the job.', 'ALWAYS : Send a notification on the completion of the job regardless of the result.', and 'NEVER : Do not send a notification for the job.'. It also mentions 'DEFAULT : Use the agent/system level job notification configuration to determine whether, and when to send the notification.'. The 'Email group' section has a dropdown menu set to '<Default>'. Below this dropdown is the text: 'Select the email-group to get the job/scheduled-task notification on completion.'. At the bottom of the dialog are three buttons: 'Cancel', 'Reset', and 'Save'.

**Create - Agent Job**

General Steps Schedules **Notifications** SQL

Send the notifications **ALWAYS**

Determines when to send a notification for the job:

**ON FAILURE :**  
Send a notification on the failure/interruption of the job.

**ALWAYS :**  
Send a notification on the completion of the job regardless of the result.

**NEVER :**  
Do not send a notification for the job.

**DEFAULT :**  
Use the agent/system level job notification configuration to determine whether, and when to send the notification.

Email group **<Default>**

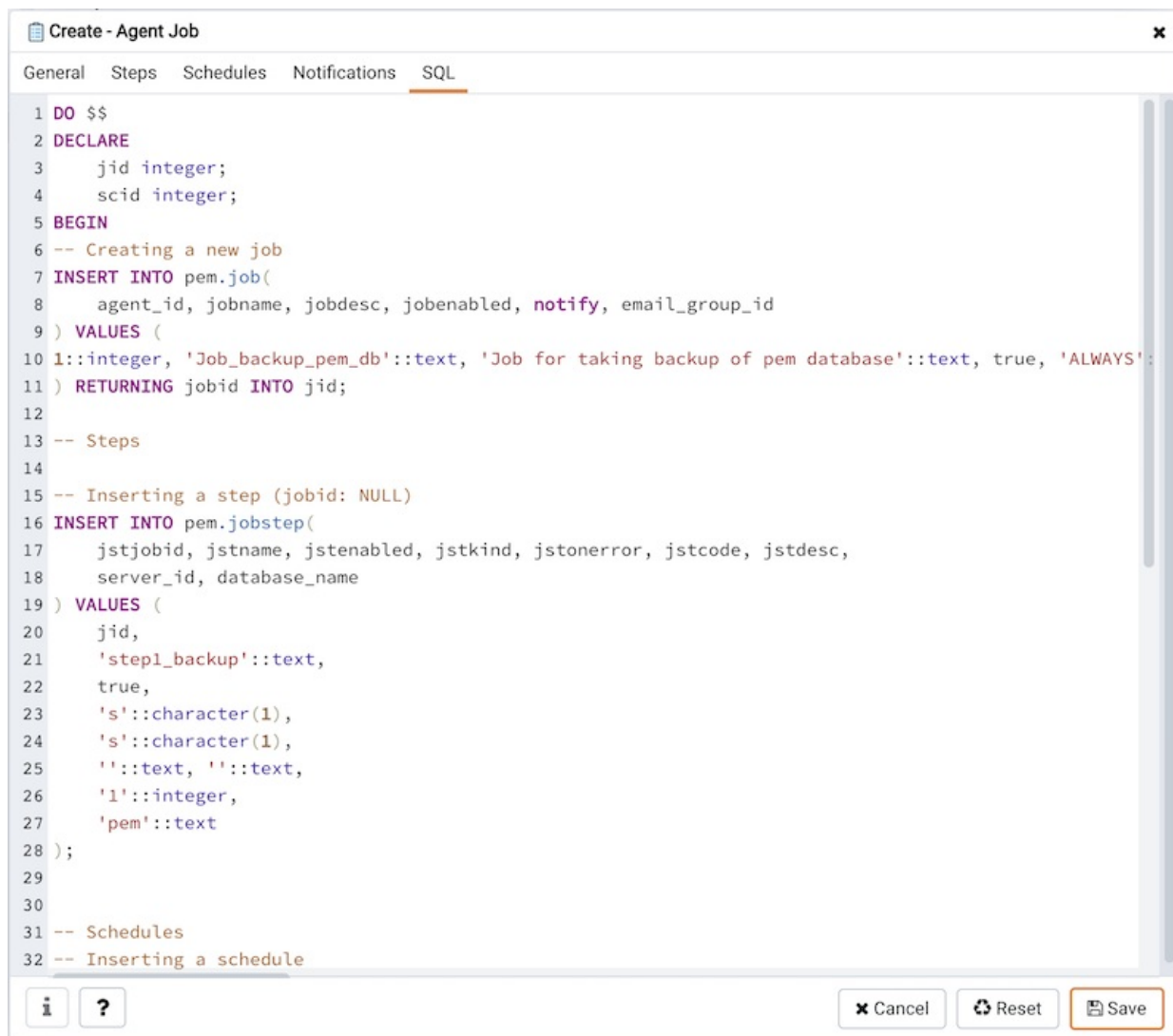
Select the email-group to get the job/scheduled-task notification on completion.

**Cancel Reset Save**

Use the fields on the **Notifications** tab to configure the email notification settings for a job:

- Use the **Send the notifications** field to specify when you want the email notifications to be sent.
- Use the **Email group** field to specify the email group that should receive the email notification.

When you've finished defining the schedule, you can use the **SQL** tab to review the code that will create or modify your job.



Click the **Save** button to save the job definition, or **Cancel** to exit the job without saving. Use the **Reset** button to remove your unsaved entries from the dialog.

After saving a job, the job will be listed under the **Jobs** node of the PEM tree control of the server on which it was defined. The **Properties** tab in the PEM console will display a high-level overview of the selected job, and the Statistics tab will show the details of each run of the job. To modify an existing job or to review detailed information about a job, right-click on a job name, and select **Properties** from the context menu.

### 34.4.17 Sending email notifications for a job

You can configure the settings in PEM console for sending the SMTP trap on success or failure of a system-generated job (listed under scheduled tasks) or a custom-defined agent job. For information on custom-defined agent job, see 'Creating PEM Scheduled Jobs'. These email notification settings can be configured at following three levels (in order of precedence) to send email notifications to the specified user group:

- Job level
- Agent level
- PEM server level (default level)

#### Configuring job notifications at job level

You can configure email notification settings at job level only for a custom-defined agent job in one of the following ways:



- For a new agent job, you can configure the email notification settings in the **Notification** tab of **Create-Agent Job** wizard while creating the job itself.
- For an existing custom-defined job, you can edit the properties of the job and configure the notification settings.

**Create - Agent Job**

General Steps Schedules **Notifications** SQL

Send the notifications **ALWAYS**

Determines when to send a notification for the job:

**ON FAILURE :**  
Send a notification on the failure/interruption of the job.

**ALWAYS :**  
Send a notification on the completion of the job regardless of the result.

**NEVER :**  
Do not send a notification for the job.

**DEFAULT :**  
Use the agent/system level job notification configuration to determine whether, and when to send the notification.

Email group **<Default>**

Select the email-group to get the job/scheduled-task notification on completion.

Cancel Reset Save

Use the fields on the Notifications tab to configure the email notification settings on job level:

- Use the **Send the notifications** field to specify when you want the email notifications to be sent.
- Use the **Email group** field to specify the email group that should receive the email notification.

### Configuring job notifications at agent level

Select the agent in the tree view, right click and select **Properties** . In the Properties dialog, select the **Job notifications** tab.

**Postgres Enterprise Manager Host (1)**

General **Job Notifications** Agent Configurations

Override default configuration? ☐ No  
 Select to override the default configuration for job notifications. If selected, the following settings will determine whether, when, and which email group will receive the job notification for this agent.

Email on job completion? ☐ No  
 Select to receive a notification email on completion of a job (regardless of the result) of this agent.

Email on a job failure? ☐ No  
 Select to receive a notification email only on failure of a job of this agent.

Email group   
 Select the email-group that will receive the notification on completion of a job or scheduled task.

Cancel Reset Save

Use the fields on the Job notifications tab to configure the email notification settings on agent level:

- Use the **Override default configuration?** switch to specify if you want the agent level job notification settings to override the default job notification settings. If you select Yes for this switch, you can use the rest of the settings on this dialog to define when and to whom the job notifications should be sent. Please note that the rest of the settings on this dialog work only if you enable the **Override default configuration?** switch.
- Use the **Email on job completion?** switch to specify if the job notification should be sent on the successful job completion.
- Use the **Email on a job failure?** switch to specify if the job notification should be sent on the failure of a job.
- Use the **Email group** field to specify the email group to whom the job notification should be sent.

### Configuring job notifications at server level

You can use the **Server Configuration** dialog to provide information about your email notification configuration at PEM server level. To open Server Configuration dialog, select **Server Configuration...** from the PEM client's Management menu.

**Server Configuration**

Search by parameter name

|                                  |  |         |
|----------------------------------|--|---------|
| job_failure_notification         | <input type="checkbox"/> False           | t/f     |
| job_notification_email_group     | default                                  |         |
| job_retention_time               | 30                                       | days    |
| job_status_change_notification   | <input checked="" type="checkbox"/> True | t/f     |
| long_running_transaction_minutes | 5  | minutes |
| max_metrics_per_group_chart      | 16                                       |         |
| nagios_cmd_file_name             | /usr/local/nagios/var/rw/nagios.cmd      |         |
| nagios_enabled                   | <input checked="" type="checkbox"/> True | t/f     |
| nagios_medium_alert_as_critical  | <input type="checkbox"/> False           | t/f     |
| nagios_spool_retention_time      | 7  | days    |
| probe_log_retention_time         | 30                                       | days    |
| reminder_notification_interval   | 24                                       | hours   |

? Cancel Reset Save

Four server configuration parameters specify information about your job notification preferences at PEM server level:

- Use the `job_failure_notification` switch to specify if you want to send email notification after each job failure.
- Use the `job_notification_email_group` parameter to specify the email group that should receive the email notification.
- Use the `job_retention_time` parameter to specify the number of days that non-recurring scheduled tasks should be retained in the system.
- Use the `job_status_change_notification` switch to specify if you want to send email notification after each job status change, irrespective of its status being a failure, success, or interrupted.

## 34.4.18 Task Viewer

Postgres Enterprise Manager runs tasks on managed servers by scheduling them on the PEM server for the agent on the managed server to execute at the appropriate time. Tasks may be one-off, or recurring and are comprised of one or more steps, which may be a SQL script, a batch/shell script, or an internal function in the PEM agent. Tasks may be viewed using the *Scheduled Tasks dialogue*.

To open the *Scheduled Tasks dialogue*, select either a PEM Agent or a managed server in the tree control of the PEM client and select the *Scheduled Tasks* menu option from the *Server* sub-menu of the *Management* menu, or from the context menu.

**Tasks**

Manage Tasks Show system tasks? ☐ No

|  | Logs | Status | Enabled?                                 | Name                                  | Agent                            | Owner    |
|--|------|--------|--|---------------------------------------|----------------------------------|----------|
|  |      |        | <input checked="" type="checkbox"/> True | PEM Log Manager Log Import - Server 1 | Postgres Enterprise Manager Host | postgres |

**General**

|             |   |
|-------------|---|
| Status      | Successfully finished                   |
| Enabled?    | <input checked="" type="checkbox"/> Yes |
| Owner       | postgres                                |
| Name        | PEM Log Manager Log Import - Server 1   |
| Agent       | Postgres Enterprise Manager Host        |
| Description | Server ID: 1, agent ID: 1               |
| Last run    | 2020-04-28 15:05:03                     |
| Next run    | 2020-04-28 15:10:00                     |
| Created     | 2020-04-27 15:46:59                     |

The dialogue displays the task data relating to the selected object when it was opened. The following details are shown:

- **Status** - The status of the task following the last execution.
- **Enabled?** - An indicator showing whether the task is enabled or not.
- **Name** - The name of the task.
- **Server** - The server on which the task will be executed, where applicable.
- **Description** - A description of the task.
- **Last Run** - The time the task was last executed.
- **Next Run** - The time the task is next scheduled to execute, if any.
- **Created** - The time and date that the task was created.

In order to [view the log records](#) for a task, select it in the list and click the [Log Viewer](#) button.

In order to remove tasks and their associated log records (if present), click the checkbox to select each task to be removed, and then click the [Remove](#) button.

**Note:** Tasks with no next run date will automatically be removed from the PEM server when the last run date is more than [probe\\_log\\_retention\\_time](#) days ago.

Contents:

34.4.18.1 Log Viewer

When PEM executes [scheduled tasks](#), log records are created to record the status of each step of the task for diagnostic purposes. Log records can be viewed on the *Log Viewer* dialogue, opened from the [Scheduled Task dialogue](#).

Tasks

Manage Tasks

Show system tasks?

No

|  | Logs | Status                 | Enabled? | Name                                  |        |        |                     | Agent                            |        | Owner       |
|--|------|------------------------|----------|---------------------------------------|--------|--------|---------------------|----------------------------------|--------|-------------|
|  |      |                        |          | PEM Log Manager Log Import - Server 1 |        |        |                     | Postgres Enterprise Manager Host |        | postgres    |
|  |      |                        |          | Server log configuration request      |        |        |                     | Postgres Enterprise Manager Host |        | postgres    |
|  |      |                        |          | Server log configuration request      |        |        |                     | Postgres Enterprise Manager Host |        | postgres    |
|  |      | Steps                  |          | Execution date                        |        |        |                     | Description                      |        |             |
|  |      |                        |          | 2020-04-27                            |        |        |                     | Server ID: 1, agent ID: 1        |        |             |
|  |      | Step                   |          | Type                                  | Status | Result | Start / Next run    | Duration                         | Output | Log details |
|  |      | Modify postgresql.conf |          | Internal                              |        | 0      | 2020-04-27 14:32:49 | 00:00:00.223211                  |        |             |
|  |      | Server Restart         |          | Internal                              |        | 0      | 2020-04-27 14:32:49 | 00:00:13.287379                  |        |             |

The dialogue displays the log data relating to each step of the task:

- **Step** - The name of the step.
- **Type** - The type of the step, one of SQL, Batch or Internal.
- **Status** - The status of the step.
- **Result** - The numeric result of the step. For a batch step, this will be the return code of the script.
- **Start / Next Run** - The schedule for the next run.
- **Duration** - The duration of the step.
- **Output** - The output text from the step, if any.

34.4.19 Monitoring a Failover Manager Cluster

You can configure PEM to display status information about one or more Failover Manager clusters on the Streaming Replication dashboard. Before

configuring PEM to monitor a Failover Manager cluster, you must install and configure Streaming Replication and Failover Manager on the cluster.

Please note that your Streaming Replication `standby.signal` file must include the following parameters:

- `primary_conninfo`
- `promote_trigger_file`

For information about installing and configuring Failover Manager and Streaming Replication, please see [Failover Manager docs](#).

To configure PEM to monitor a Failover Manager cluster, use the PEM client to create a server definition for the primary node of the Failover Manager cluster. Use the tabs on the [New Server Registration](#) dialog to specify general connection properties for the primary node; use fields on the [Advanced](#) tab to specify information about the Failover Manager cluster:

- Use the [EFM Cluster Name](#) field to specify the name of the Failover Manager cluster. The cluster name is the prefix of the name of the cluster properties file. For example, if your cluster properties file is named `efm.properties`, your cluster name is `efm`.
- Use the [EFM Installation Path](#) field to specify the location of the Failover Manager binary file. By default, the Failover Manager binary file is installed in `/usr/efm-x.x/bin`.

After saving the server definition, the primary node will be included in the list of servers under the [PEM Server Directory](#) in the PEM client [Object browser](#) tree, and will be displayed on the [Global Overview](#) dashboard.

To include Failover Manager information on the Streaming Analysis dashboard, you must enable the following probes for each node in the Failover Manager cluster:

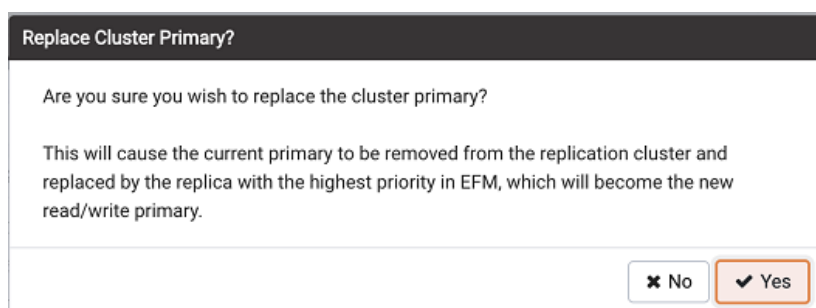
- Failover Manager Cluster Info
- Failover Manager Node Status

To enable a probe, right click on the node name, and select [Manage Probes](#) from the [Management](#) menu.

To view the [Streaming Replication Analysis](#) dashboard and the status of the Failover Manager cluster, right click on the name of the primary node in the [Object browser](#) tree control and navigate through the [Dashboards](#) menu to select [Streaming Replication Analysis](#).

## Promoting a Cluster

Select the [Replace Cluster Primary](#) from [Server](#) under the [Tools](#) menu to start the failover process. When you select [Replace Cluster Primary](#), a popup opens, asking you to confirm that you wish to replace the current primary node:



Select [No](#) to exit the popup without replacing the current primary node.

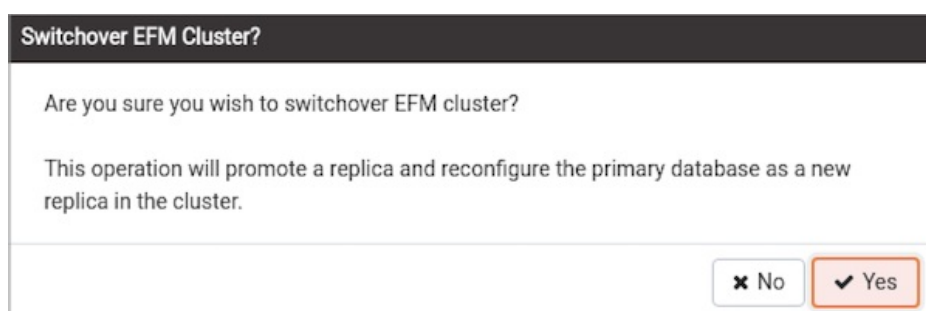
Select [Yes](#) to remove the current primary node from the Failover Manager cluster and promote a replica node to the role of read/write primary node within a Failover Manager cluster. The node with the highest promotion priority (defined in Failover Manager) will become the new primary node. PEM will display a dialog, reporting the job status.



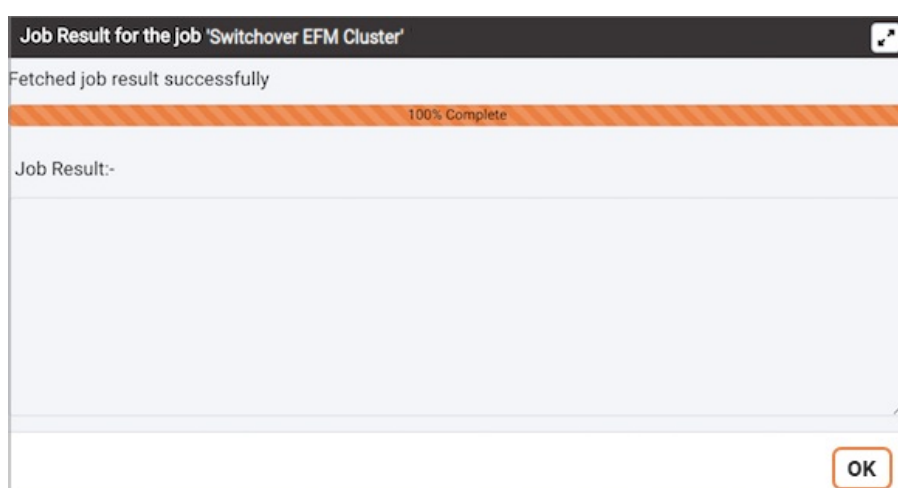
When the job completes and the Streaming Replication Analysis dashboard refreshes, you can review the **Failover Manager Node Status** table to confirm that a replica node has been promoted to the role of primary within the Failover Manager cluster.

### Switchover EFM Cluster

You can use the PEM client to switchover the primary node of a Failover Manager cluster with a replica node. To initiate the switchover process, select Switchover EFM Cluster from the Tools menu. A dialog opens, asking you to confirm that you wish to switchover EFM cluster.



Select **Yes** to switchover EFM cluster from the Failover Manager cluster and promote a replica node to the role of read/write primary node and reconfigure the primary database as a new replica within a Failover Manager cluster. The node with the highest promotion priority (defined in Failover Manager) will become the new primary node. PEM will display a dialog, reporting the job status.



When the job completes and the Streaming Replication Analysis dashboard refreshes, you can review the **Failover Manager Node Status** table to confirm that a switchover within the Failover Manager cluster.

## 34.4.20 Monitoring a PGD Group

PGD (Bi-Directional Replication) is a PostgreSQL extension that provides a multi-master replication between the Postgres clusters. Before you monitor PGD through PEM console, you must ensure that your PGD nodes are up and running. For more information about PGD please visit the [EDB Doc Website](#).

You can configure PEM to display status information about one or more PGD Nodes via dashboards from PEM 8.1.0 and PGD 3.7.9 onwards.

To configure PEM to monitor **PGD Nodes**, use the PEM Web Client to create a server definition. Use the tabs on the **New Server Registration** dialog to specify general connection properties for the PGD Node with the following exceptions:

- Specify the PGD-enabled database name in the **Database** field of the **PEM Agent** tab.
- Specify the user having **pgd\_monitor** or **pgd\_superuser** role in the **username** field of the **PEM Agent** tab.

After saving the server definition, the PGD Node will be included in the list of servers under the PEM Server Directory in the PEM client object browser tree. You can monitor the **PGD Nodes** from the **PGD dashboards**.

To include monitoring information on the PGD dashboards, you must enable the relative probes for each PGD Group. You can see the complete list of PGD Probes [here](#).

To enable a probe, right click on the node name, and select **Manage Probes** from the **Management** menu.

To monitor the PGD Node, right click on the name of the PGD Node in the Object browser tree control and navigate through the Dashboards menu to select **PGD Admin** or **PGD Group Monitoring** or **PGD Node Monitoring** dashboard.

## 34.4.21 Performance Diagnostic

You can use the Performance Diagnostic dashboard to analyze the database performance for Postgres instances by monitoring the wait events. To display the diagnostic graphs, PEM uses the data collected by EDB Wait States module.

Performance Diagnostic feature is supported for Advanced Server databases from PEM 7.6 version onwards and for PostgreSQL databases it is supported from PEM 8.0 onwards.

### Note

For PostgreSQL databases, Performance Diagnostics is supported only for versions 10, 11, 12, and 13 installed on supported platforms.

For more information on EDB Wait States, see [EDB wait states docs](#).

You can analyze the Wait States data on multiple levels by narrowing down your selection of data. Each level of the graph is populated on the basis of your selection of data at the higher level.

Prerequisite:

- For PostgreSQL, you need to install **edb\_wait\_states\_<X>** package from **edb.repo** where **<X>** is the version of PostgreSQL Server. You can refer to [EDB Build Repository](#) for the steps to install this package. For Advanced Server, you need to install **edb-as<X>-server-edb-modules**, Where **<X>** is the version of Advanced Server.
- Once you ensure that EDB Wait States module of EDB Postgres Advanced Server is installed, then configure the list of libraries in the **postgresql.conf** file as below:

```
shared_preload_libraries = '$libdir/edb_wait_states'
```

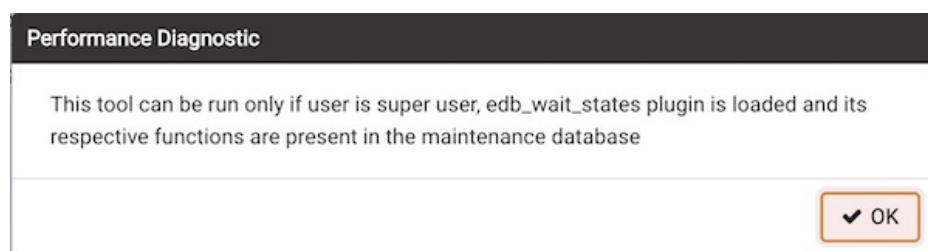
Restart the database server, and then create the following extension in the maintenance database:

```
CREATE EXTENSION edb_wait_states;
```



- You must have super user privileges to access the Performance Diagnostic dashboard.

You get the following error while accessing the Performance Diagnostic dashboard if the above prerequisites are not met:



To open the Performance Diagnostic dashboard, select **Server** and then **Performance Diagnostic...** from the **Tools** menu of the PEM client.

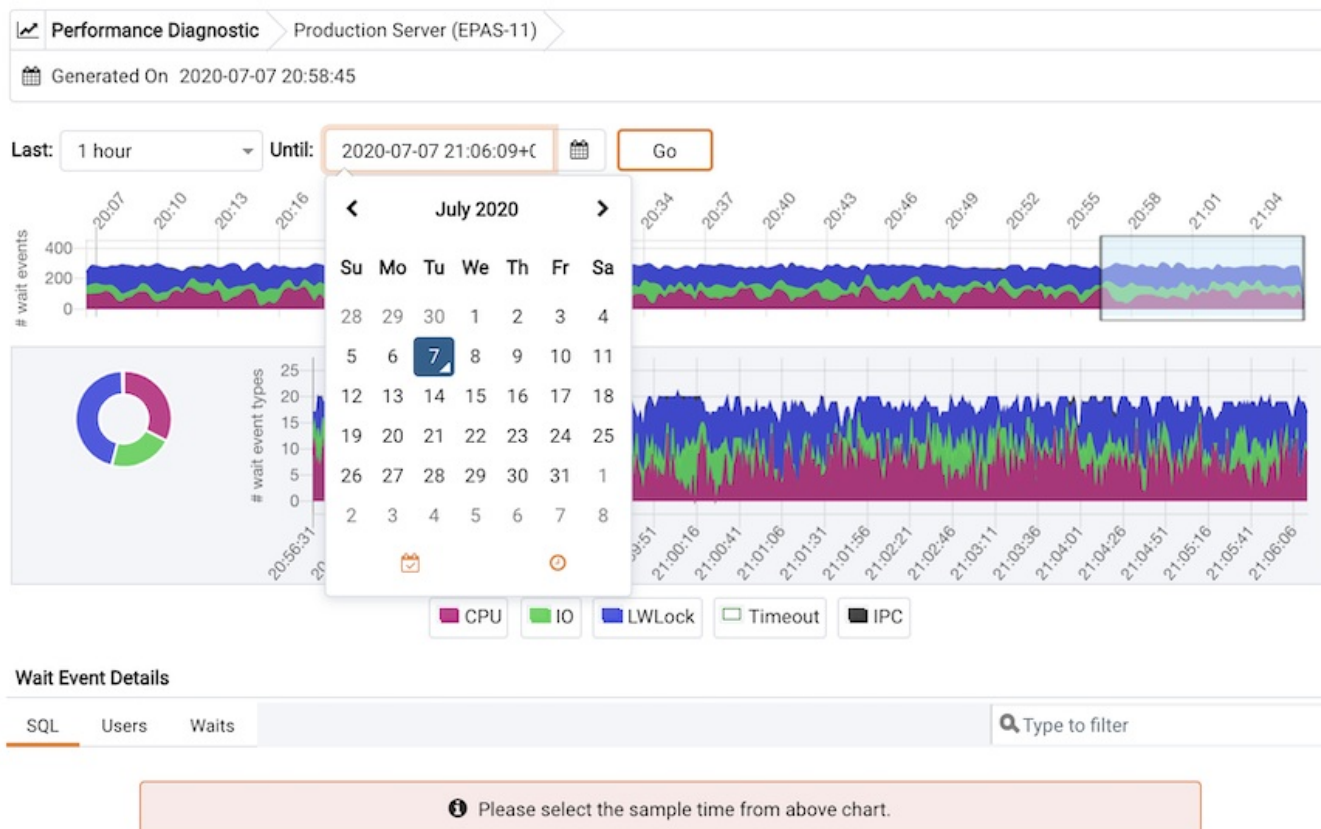


By default, the top most Performance Diagnostic graph pulls the data of last one hour, starting from current date and time. This graph shows the time series containing the number of active sessions. Each point of this time series represents the active sessions and wait events at a particular time and last 15 seconds. These sessions may or may not be waiting for an wait event, or using the CPU at a particular point in time. This time series is generated based on the wait event samples collected by the `edb_wait_states` extension.

You can also use the **Preferences** dialog to display Performance Diagnostic in a new browser tab. Use **Open in New Browser Tab?** to display the Performance Diagnostics dashboard in a new browser tab.

The range selection in the first graph is 10 minutes. You can use the **Last** drop-down list box to select the duration for which you want to see the graph: select the last 1 hour, last 4 hours, last 12 hours, or last 24 hours. You can also select the date and time through which you want the data to be displayed.





The first graph displays the number of active sessions (and - wait event types) for the selected time interval. You can narrow down the timeline in the first graph to analyze the data for a specific time period.

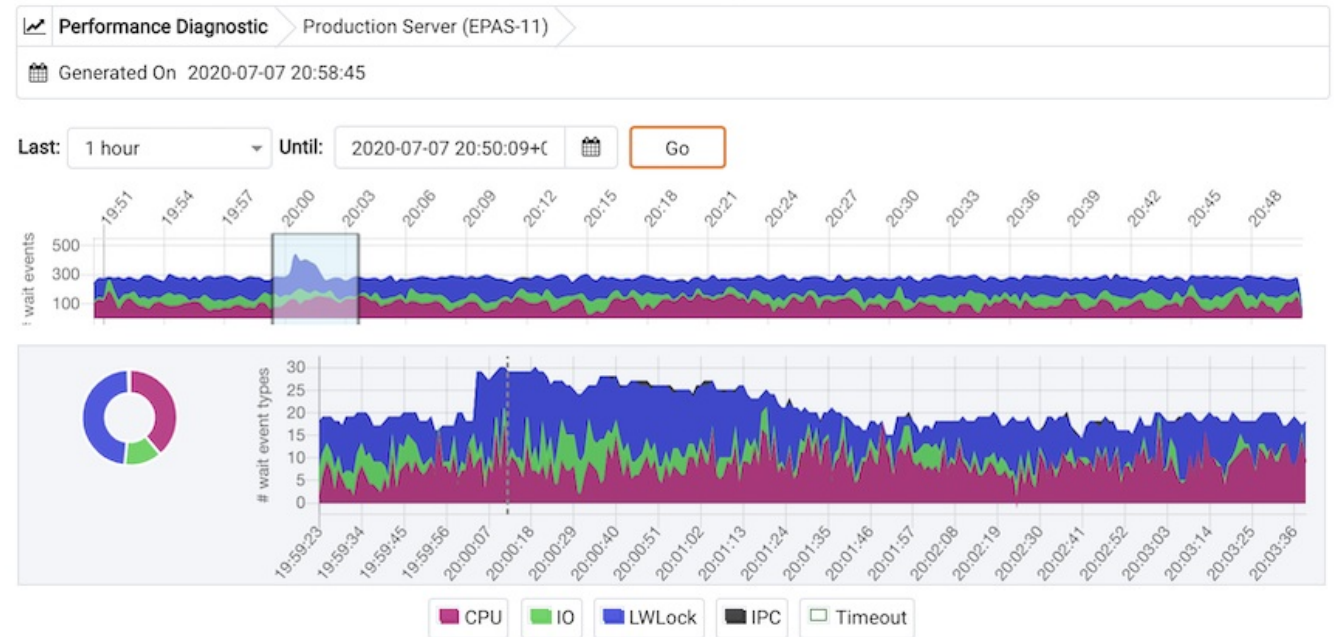
Next section plots the following graphs based on the selected time interval in the first graph:

1. Donut graph - It shows total wait event types according to the time range selection in the first graph. It helps you understand how much time was spent by those session on waiting for an event.
2. Line graph - It plots a time series with each point representing the active sessions for each sample time.

To differentiate each wait event types and the CPU usage clearly, the graph for each wait event type is displayed in a different color.

Select a particular time on the **Line graph** for which you wish to analyze the wait events; the third section displays the wait event details in the Performance Diagnostics dashboard on the basis of your selected particular time in the second graph. The third section displays wait event details on three tabs:

- The **SQL** tab displays the list of SQL queries having wait events for the selected sample time.
- The **Users** tab displays the details of the wait events grouped by users for selected sample time.
- The **Waits** tab displays the number of wait events belonging to each wait event type for the selected sample time.

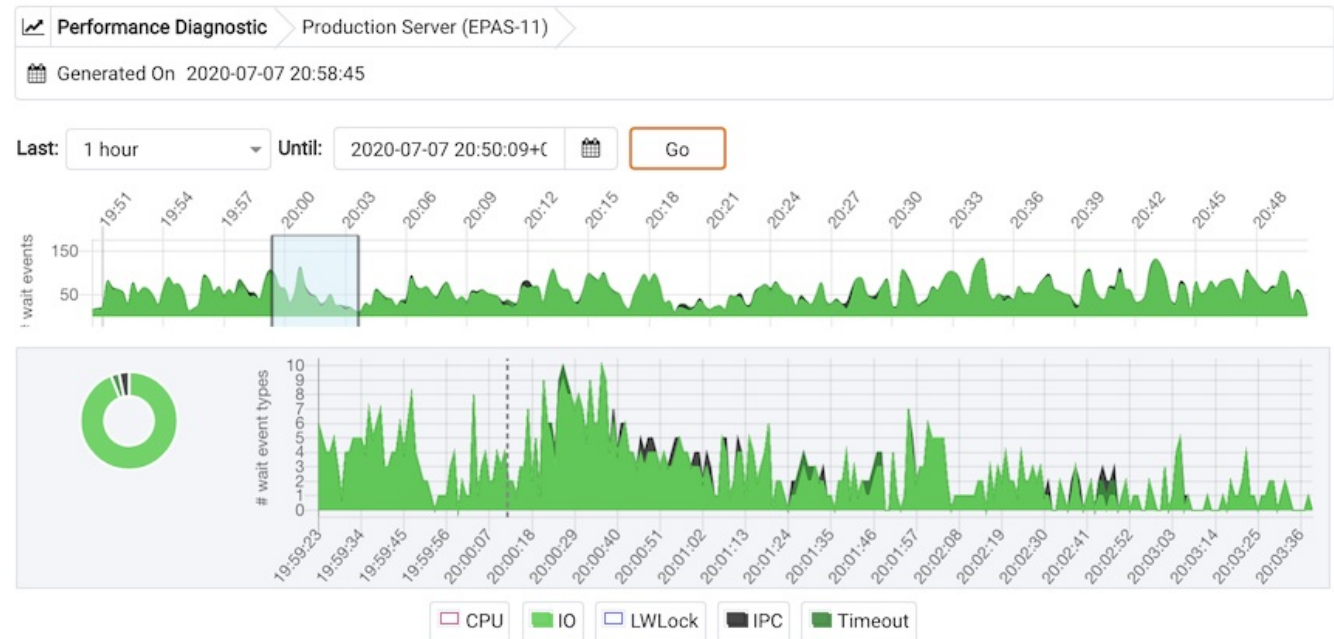


Sample time: 2020-07-07 20:00:10 +05:30

Wait Event Details

| SQL             | Users  | Waits              | Type to filter |
|-----------------|--|--------------------|----------------|
| Load By Waits ▾ | SQL  | Number of sessions |                |
|                 | <code>select l_returnflag, l_linestatus, sum(l_quantity) as</code> | 4                  |                |
|                 | <code>select c_custkey, c_name, sum(l_extendedprice * (\$1</code>  | 3                  |                |
|                 | <code>select s_name, count(*) as numwait from supplier, li</code>  | 3                  |                |
|                 | <code>select s_suppkey, s_name, s_address, s_phone, total_</code>  | 2                  |                |
|                 | <code>select o_orderpriority, count(*) as order_count from</code>  | 2                  |                |

You can click on the graph legends to show or hide a particular wait event type in all the graphs. This will make the analysis of a specific wait event type easier.



You can filter the data displayed in the rows under all the three tabs. You can also sort the data alphabetically by clicking on the column headers.

SQL tab

Wait Event Details

| SQL           | Users | Waits   |                    | Q year | X |
|---------------|-------|---|--------------------|--------|---|
| Load By Waits | SQL   |   | Number of sessions |        |   |
|               |       | <code>select nation, o_year, sum(amount) as sum_profit from</code>  |                    | 2      |   |
|               |       | <code>select o_year, sum(case when nation = \$1 then volume</code>  |                    | 1      |   |
|               |       | <code>select l_shipmode, sum(case when o_orderpriority = \$1</code> |                    | 1      |   |
|               |       | <code>select supp_nation, cust_nation, l_year, sum(volume)</code>   |                    | 1      |   |

Users tab

Wait Event Details

| SQL           | Users        | Waits            |  | Q Type to filter |
|---------------|--------------|------------------|--|------------------|
| Load By Waits | Users        | Number of Events |  | Execution Count  |
|               | enterprisedb | 6                |  | 6                |
|               | test2        | 6                |  | 6                |
|               | test1        | 5                |  | 6                |
|               | test3        | 5                |  | 7                |
|               | test4        | 3                |  | 5                |

Waits tab

Wait Event Details

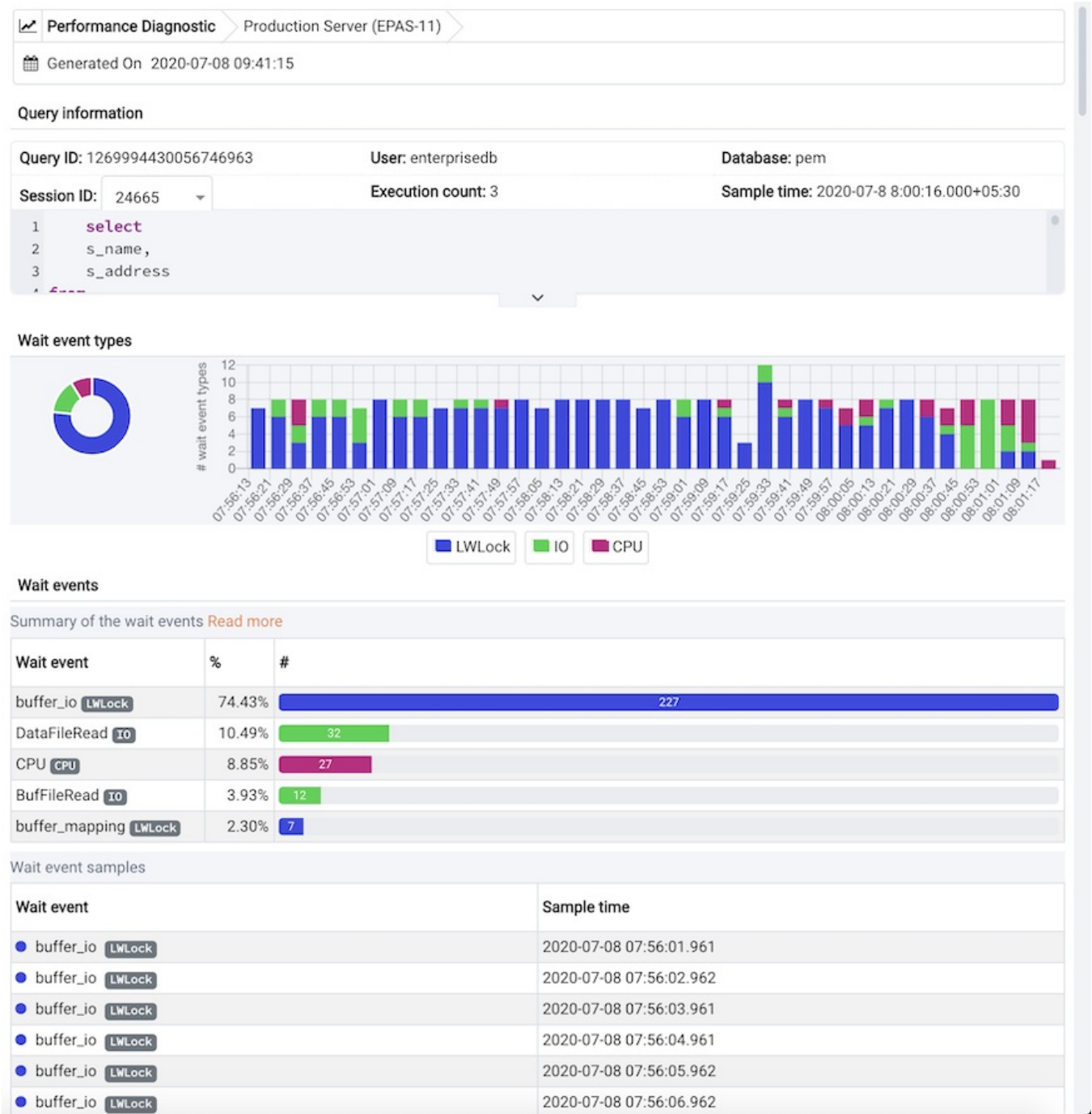
| SQL          | Users           | Waits          |                  | Q Type to filter |
|--------------|-----------------|----------------|------------------|------------------|
| Load By Wait | Wait Event Type | Wait Event     | Number of Events |                  |
|              | LWLock          | buffer_mapping | 21               |                  |
|              | IO              | DataFileRead   | 3                |                  |
|              | LWLock          | buffer_io      | 1                |                  |

Click on the Eye icon in any row of the SQL tab to display a new tab with details of the query of that particular row. This page displays Query ID and its corresponding session IDs in a dropdown list at that particular selected sample time in the Query information section. You can select the session ID for the selected query for which you want to analyze the data. You will see the details corresponding to the selected session ID and query ID. The Query information table also displays the SQL query. If the SQL query is being displayed partially, click the down arrow at the bottom of the section to view the complete SQL query.

The **Wait event types** section displays the total number of wait event types for the selected session ID and query ID. It shows two type of graphs:

- 1. Donut graph - It shows the proportions of categorical data, with the size of each piece representing the proportion of each wait event type.
- 2. Timeline bar graph - It can be used to visualize trends in counts of wait event types over time.

To differentiate clearly, each wait event type is represented by a different color in the bar graph.



The **Wait events** section has a table displaying all the wait events occurred during the query execution. It displays data in decreasing order by number of the wait events. Second table displays the wait event with sample time occurred over the period of whole query execution. It allows to analyze the wait events during the query execution over the period of time. It shows the actual samples collected by the EDB Wait States extension for that particular query ID and session ID.

34.4.22 Reports

You can generate the System Configuration report and Core Usage report for all locally and remotely managed servers. To generate this report, select **Reports** from the **Management** Menu.

Reports has following options:

- System Configuration Report (JSON)
- System Configuration Report (HTML)

- Core Usage Report (JSON)
- Core Usage Report (HTML)

Please note that only superusers or the users with the pem\_admin role permission can download the System Configuration or Core Usage reports.

Also note that information in these reports will reflect the latest probe run time.

## System Configuration Report

The System Configuration Report provides detailed information about the PEM Agents group, PEM Server directory group and custom groups listed under browser tree. These groups can contain Postgres Enterprise Manager, PEM Agents and Database servers. You can download this report in HTML as well as in JSON format.

The **Postgres Enterprise Manager Summary** provides details about:

- The Postgres Enterprise Manager backend database server version
- Application Version
- User name accessing the application
- Python version
- Flask version
- Platform specific information

The **Summary**, details about the number of agents and servers are provided.

System Configuration Report

Generated On: 2020-04-28 14:30:49

Go to: PEM Agents

Postgres Enterprise Manager Summary

| Parameter       | Value   |
|-----------------|---|
| Name            | Postgres Enterprise Manager   |
| Backend version | PostgreSQL 12.1 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-39), 64-bit   |
| App version     | 7.14.0-dev (schema: 202003031)  |
| User            | postgres  |
| Python version  | 3.7.5   |
| Flask version   | 1.0.4   |
| Platform        | <div>System : Darwin<br/>Node : Laptop358.pn.in<br/>Release : 18.7.0<br/>Version : Darwin Kernel Version 18.7.0: Thu Jan 23 06:52:12 PST 2020; root:xnu-4903.278.25~1/RELEASE_ARM64_T8020<br/>Machine : x86_64<br/>Processor : i386</div> |

Summary

| Parameter | Value   |
|-----------|---|
| Agents    | <div>Windows : 0<br/>Linux : 2</div>  |
| Servers   | <div>PG : 2<br/>EPAS : 2<br/>Unknown : 0<br/>Locally Managed : 3<br/>Remotely Managed : 1<br/>Unmanaged : 0</div> |

Group: PEM Agents

Group: PEM Server Directory

The **Group: PEM Agents**, details about PEM Agent, CPU, Disk Utilization as well as Memory details are provided.

System Configuration Report

Generated On: 2020-04-28 14:30:49

Go to: PEM Agents

> Postgres Enterprise Manager Summary

> Summary

> Group: PEM Agents

Agent: Postgres Enterprise Manager Host

> Agent Details

| Parameter            | Value  |
|----------------------|--|
| Platform             | Linux-x64  |
| OS                   | CentOS Linux release 7.5.1804 (Core)   |
| Version              | 7.14.0-dev   |
| Active               | True   |
| Hostname             | localhost.localdomain  |
| Domain Name          | (none)   |
| Bound Local Servers  | <ul style="list-style-type: none"><li>Postgres Enterprise Manager Server</li><li>EDB Postgres Advanced Server 11</li><li>EPAS_12</li></ul> |
| Bound Remote Servers | (none)   |

> CPU

Total CPU Cores: 2

Average CPU Utilization (%): 25.31

| Core ID | Load Percentage |
|---------|-----------------|
| CPU0    | 25.267327       |
| CPU1    | 25.353135       |

> Disk Utilization

Total Disk Size (MB): 32098

Disk Space Used (MB): 13791

Disk Space Available (MB): 16669

Disk Utilization (%): 42.97

| Mount Point | File System | Size (MB) | Space Used (MB) | Space Available (MB) |
|-------------|-------------|-----------|-----------------|----------------------|
| /           | /dev/sda3   | 31622     | 13657           | 16352                |
| /boot       | /dev/sda1   | 476       | 134             | 317                  |

> Memory Details

| Parameter               | Value |
|-------------------------|-------|
| Free RAM (MB)           | 1050  |
| Memory Usage Percentage | 72.17 |
| Total Swap Memory (MB)  | 7999  |
| Free Swam Memory (MB)   | 6589  |
| Swap Usage Percentage   | 17.63 |

Agent: localhost.localdomain

> Agent Details

> CPU

> Disk Utilization

> Memory Details

> Group: PEM Server Directory

The **Group: PEM Server Directory** , provides details about:

- The database server version
- Host
- Port
- Database name
- Database size
- Tablespace size

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

592



System Configuration Report

Generated On: 2020-04-28 14:30:49

Go to: PEM Agents

Postgres Enterprise Manager Summary

Summary

Group: PEM Agents

Group: PEM Server Directory

Server: Postgres Enterprise Manager Server

Server Details

| Parameter         | Value   |
|-------------------|---|
| Agent             | Postgres Enterprise Manager Host  |
| Host              | 192.168.1.19  |
| Port              | 5432  |
| Database          | postgres  |
| Version           | PostgreSQL 12.1 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-39), 64-bit |
| Service Id        | postgresql-12   |
| Remote Monitored? | False   |
| Active            | True  |

Database Details

| Name          | Size (MB) | Tablespace Name |
|---------------|-----------|-----------------|
| postgres      | 8         | pg_default      |
| edbstore_temp | 8         | pg_default      |
| hr            | 8         | pg_default      |
| pem           | 2407      | pg_default      |
| testdb        | 27        | pg_default      |
| db01          | 8         | pg_default      |

Tablespace Details

| Name       | Size (MB) |
|------------|-----------|
| pg_global  | 0         |
| pg_default | 2482      |

Server: EPAS\_12

Server Details

Database Details

Tablespace Details

Server: EDB Postgres Advanced Server 11

Server Details

Database Details

Tablespace Details

Server: PGSQL12\_Centos7\_1

Server Details

Database Details

Tablespace Details

Please note that here Group Server Name depends on the group name to which the server is added.

Core Usage Report


The Core Usage report provides detailed information about number of cores specific to:

- The server type
- Database version
- Platform and group name

Also, gives detailed information about locally managed servers with:

- Type
- Host
- Port
- Platform
- Cores
- RAM

Core Usage Report

Generated On: 2020-02-18 16:08:47 IST

Using: Postgres Enterprise Manager

Version: 7.13.0-dev (schema: 202001011)

Core Summary

Total Number of Cores: 13

| Server Type                  | Number of Servers | Number of Cores |
|------------------------------|-------------------|-----------------|
| EDB Postgres Advanced Server | 1                 | 4               |
| PostgreSQL                   | 2                 | 5               |
| BART                         | 1                 | 4               |

Database Version

| Database Version   | Number of Servers | Number of Cores |
|--------------------|-------------------|-----------------|
| PostgreSQL 10      | 1                 | 4               |
| PostgreSQL 11      | 1                 | 1               |
| Advanced Server 11 | 1                 | 4               |

Platform

| Platform    | Number of Servers | Number of Cores |
|-------------|-------------------|-----------------|
| Windows-x64 | 1                 | 1               |
| Linux-x64   | 3                 | 12              |

Group Name

| Group Name           | Number of Servers | Number of Cores |
|----------------------|-------------------|-----------------|
| PEM Server Directory | 3                 | 9               |

Server Core Summary

Locally Managed Servers: 3

| Name                               | Type                         | Host:Port      | Platform    | Cores | Total RAM (MB) |
|------------------------------------|------------------------------|----------------|-------------|-------|----------------|
| Windows PostgreSQL 11              | PostgreSQL                   | 127.0.0.1:5432 | Windows-x64 | 1     | 2047           |
| PostgreSQL-10-Local                | PostgreSQL                   | localhost:5432 | Linux-x64   | 4     | 5786           |
| Postgres Enterprise Manager Server | EDB Postgres Advanced Server | 127.0.0.1:5444 | Linux-x64   | 4     | 5786           |
|                                    |                              |                |             | 9     | 13619          |

Remotely Managed Servers: 1

| Name                 | Type       | Host:Port        |
|----------------------|------------|------------------|
| PostgreSQL-11-Remote | PostgreSQL | 172.19.12.3:5432 |

Unmanaged Servers: 1

| Name                           | Host:Port           |
|--------------------------------|---------------------|
| Performance Diagnostics Server | 172.16.254.22: 5444 |

### 34.5 Management Basics

PEM provides a graphical interface that you can use to simplify management of your Postgres servers and the objects that reside on them.

The Grant Wizard simplifies the task of privilege management; to open the Grant Wizard, highlight the name of a server, database, or schema in the PEM client tree control, and select **Grant Wizard...** from the **Tools** menu.

- [Grant Wizard](#)

PEM provides an easy to use environment in which to manage restore points, import/export tasks, and organize vacuum/analyze management.

- [Add named restore point Dialog](#)
- [Import/Export data Dialog](#)
- [Maintain a database object](#)
- [Storage Manager](#)

A powerful, but user-friendly interface provides an easy way to use take backups and create copies of databases or database objects.

- [Backup Dialog](#)
- [Backup Globals Dialog](#)
- [Backup Server Dialog](#)
- [Restore Dialog](#)

You can also use the client to manage objects that reside on managed and unmanaged database servers:



- [Managing Cluster Level Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)

### 34.5.1 Grant Wizard

The **Grant Wizard** tool is a graphical interface that allows you to manage the privileges of one or more database objects in a point-and-click environment. A search box, dropdown lists, and checkboxes facilitate quick selections of database objects, roles and privileges.

The wizard organizes privilege management through a sequence of windows: **Object Selection (step 1 of 3)** , **Privileges Selection (step 2 of 3)** and **Final (Review Selection) (step 3 of 3)** .The **Final (Review Selection)** window displays the SQL code generated by wizard selections.

To launch the **Grant Wizard** tool, select a database object in the **Browser** tree control, then navigate through **Tools** on the menu bar to click on the **Grant Wizard** option.

Grant Wizard - Object Selection (step 1 of 3)

Please select the objects to grant privileges to from the list below.

Search

| <input type="checkbox"/>            | Object Type | Schema | Name  |
|-------------------------------------|-------------|--------|---|
| <input checked="" type="checkbox"/> | Function    | public | akeys(hstore)                               |
| <input checked="" type="checkbox"/> | Function    | public | avals(hstore)                               |
| <input type="checkbox"/>            | Function    | public | defined(hstore, text)                       |
| <input type="checkbox"/>            | Function    | public | delete(hstore, text[])                      |
| <input type="checkbox"/>            | Function    | public | delete(hstore, hstore)                      |
| <input type="checkbox"/>            | Function    | public | delete(hstore, text)                        |
| <input type="checkbox"/>            | Function    | public | each(hs hstore, OUT key text, OUT value tex |
| <input type="checkbox"/>            | Function    | public | exist(hstore, text)                         |
| <input type="checkbox"/>            | Function    | public | exists_all(hstore, text[])                  |
| <input type="checkbox"/>            | Function    | public | exists_any(hstore, text[])                  |

?

Cancel

Back

Next

Finish

Use the fields in the **Object Selection (step 1 of 3)** window to select the object or objects on which you are modifying privileges. Use the **Search by object type or name** field to locate a database object, or use the scrollbar to scroll through the list of all accessible objects.

- Each row in the table lists object identifiers; check the checkbox in the left column to include an object as a target of the Grant Wizard. The table displays:
  - The object type in the **Object Type** field
  - The schema in which the object resides in the **Schema** field
  - The object name in the **Name** field.

Click the **Next** button to continue, or the **Cancel** button to close the wizard without modifying privileges.

Grant Wizard - Privileges Selection (step 2 of 3)

Please select privileges for the selected objects.

Privileges

| Grantee                                    | Privileges   | Grantor                                   |
|--|--|---|
| <div><div></div><div>pem_admin</div></div> | <div><div><input checked="" type="checkbox"/> EXECUTE</div><div><input type="checkbox"/> WITH GRANT OPTION</div></div> | <div><div></div><div>postgres</div></div> |

?

Cancel

Back

Next

Finish

Use the fields in the **Privileges Selection (step 2 of 3)** window to grant privileges. If you grant a privilege WITH GRANT OPTION, the Grantee will have the right to grant privileges on the object to others. If WITH GRANT OPTION is subsequently revoked, any role who received access to that object from that Grantee (directly or through a chain of grants) will lose thier privileges on the object.

- Click the **Add** icon (+) to assign a set of privileges.
- Select the name of the role from the drop-down listbox in the **Grantee** field.
- Click inside the **Privileges** field. Check the boxes to the left of one or more privileges to grant the selected privileges to the specified user. If privileges have previously been granted on a database object, unchecking a privilege for a group or user will result in revoking that privilege.
- The current user, who is the default grantor for granting the privilege, is displayed in the **Grantor** field.
- Click the **Add** icon (+) to assign a set of privileges to another role; to discard a privilege, click the trash icon to the left of the row and confirm deletion in the **Delete Row** dialog.

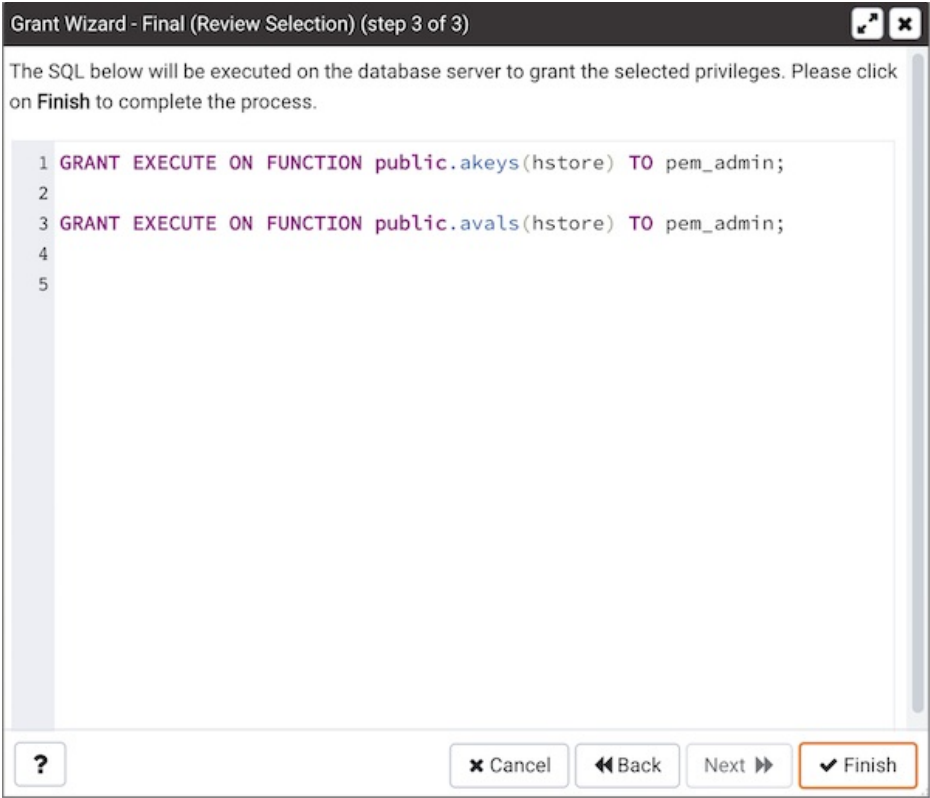
For more information about granting privileges on database objects, see the [PostgreSQL core documentation](#).

Click the **Next** button to continue, the **Back** button to select or deselect additional database objects, or the **Cancel** button to close the wizard without modifying privileges.

Your entries in the **Grant Wizard** tool generate a SQL command; you can review the command in the **Final (Review Selection) (step 3 of 3)** window (see an example below).

Example

The following is an example of the sql command generated by user selections in the **Grant Wizard** tool:

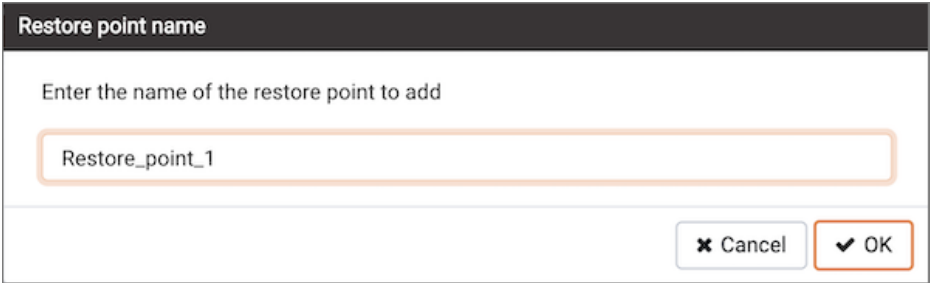


The commands displayed assign a role named `Bob` `INSERT` and `UPDATE` privileges `WITH GRANT OPTION` on the `sales_meetings` and the `sales_territories` tables.

- Click the `Back` button to select or deselect additional database objects, roles and privileges.
- Click the `Cancel` button to exit without saving work.
- Click the `Finish` button to save selections and exit the wizard.

### 34.5.2 Add named restore point Dialog

Use the `Add named restore point` dialog to take a named snapshot of the state of the server for use in a recovery file. To create a named restore point, the server's postgresql.conf file must specify a `wal_level` value of `replica` , or `logical` . You must be a database superuser to create a restore point.



When the `Restore point name` window launches, use the field `Enter the name of the restore point to add` to provide a descriptive name for the restore point.

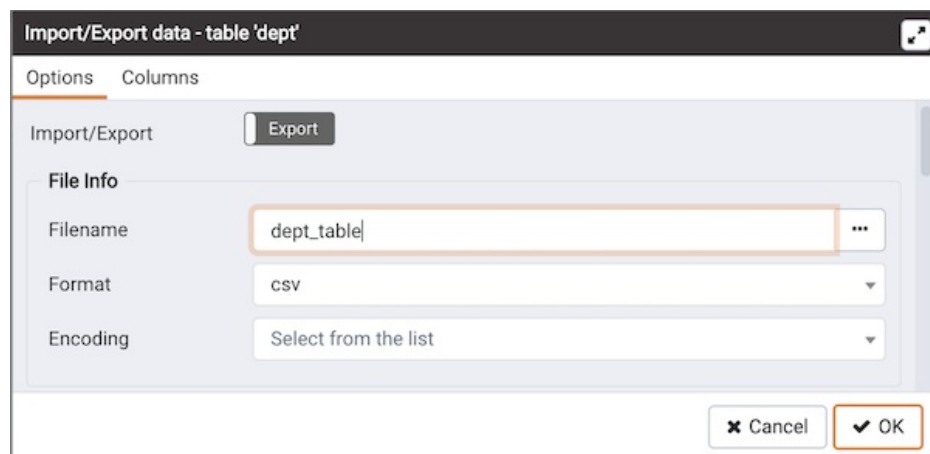
For more information about using a restore point as a recovery target, please see the [PostgreSQL documentation](#).

- Click the `OK` button to save the restore point.
- Click the `Cancel` button to exit without saving work.

### 34.5.3 Import/Export data Dialog

Use the **Import/Export data** dialog to copy data from a table to a file, or copy data from a file into a table.

The **Import/Export data** dialog organizes the import/export of data through the **Options** and **Columns** tabs.



Use the fields in the **Options** tab to specify import and export preferences:

- Move the **Import/Export** switch to the **Import** position to specify that the server should import data to a table from a file. The default is **Export**.
- Use the fields in the **File Info** field box to specify information about the source or target file:
  - Enter the name of the source or target file in the **Filename** field. Optionally, select the **Browser** icon (ellipsis) to the right to navigate into a directory and select a file.
  - Use the drop-down listbox in the **Format** field to specify the file type. Select:
    - **binary** for a .bin file.
    - **csv** for a .csv file.
    - **text** for a .txt file.
  - Use the drop-down listbox in the **Encoding** field to specify the type of character encoding.

**Import/Export data - table 'dept'**

Options Columns

**Miscellaneous**

OID

Header

Delimiter 

Specifies the character that separates columns within each row (line) of the file. The default is a tab character in text format, a comma in CSV format. This must be a single one-byte character. This option is not allowed when using binary format.

Quote 

Specifies the quoting character to be used when a data value is quoted. The default is double-quote. This must be a single one-byte character. This option is allowed only when using CSV format.

Escape 

Specifies the character that should appear before a data character that matches the QUOTE value. The default is the same as the QUOTE value (so that the quoting character is doubled if it appears in the data). This must be a single one-byte character. This option is allowed only when using CSV format.

- Use the fields in the **Miscellaneous** field box to specify additional information:
  - Move the **OID** switch to the **Yes** position to include the **OID** column. The **OID** is a system-assigned value that may not be modified. The default is **No**.
  - Move the **Header** switch to the **Yes** position to include the table header with the data rows. If you include the table header, the first row of the file will contain the column names.
  - If you are exporting data, specify the delimiter that will separate the columns within the target file in the **Delimiter** field. The separating character can be a colon, semicolon, a vertical bar, or a tab.
  - Specify a quoting character used in the **Quote** field. Quoting can be applied to string columns only (i.e. numeric columns will not be quoted) or all columns regardless of data type. The character used for quoting can be a single quote or a double quote.
  - Specify a character that should appear before a data character that matches the **QUOTE** value in the **Escape** field.

Click the **Columns** tab to continue.

**Import/Export data - table 'dept'**

Options Columns

Columns to export

An optional list of columns to be copied. If no column list is specified, all columns of the table will be copied.

NULL Strings 

Specifies the string that represents a null value. The default is \N (backslash-N) in text format, and an unquoted empty string in CSV format. You might prefer an empty string even in text format for cases where you don't want to distinguish nulls from empty strings. This option is not allowed when using binary format.

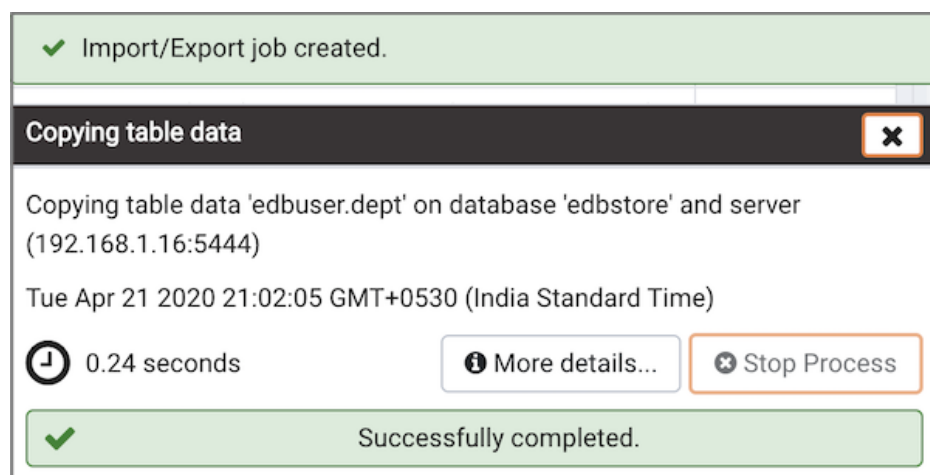
Not null columns 

Do not match the specified column values against the null string. In the default case where the null string is empty, this means that empty values will be read as zero-length strings rather than nulls, even when they are not quoted. This option is allowed only in import, and only when using CSV format.

Use the fields in the **Columns** tab to select the columns that will be imported or exported:

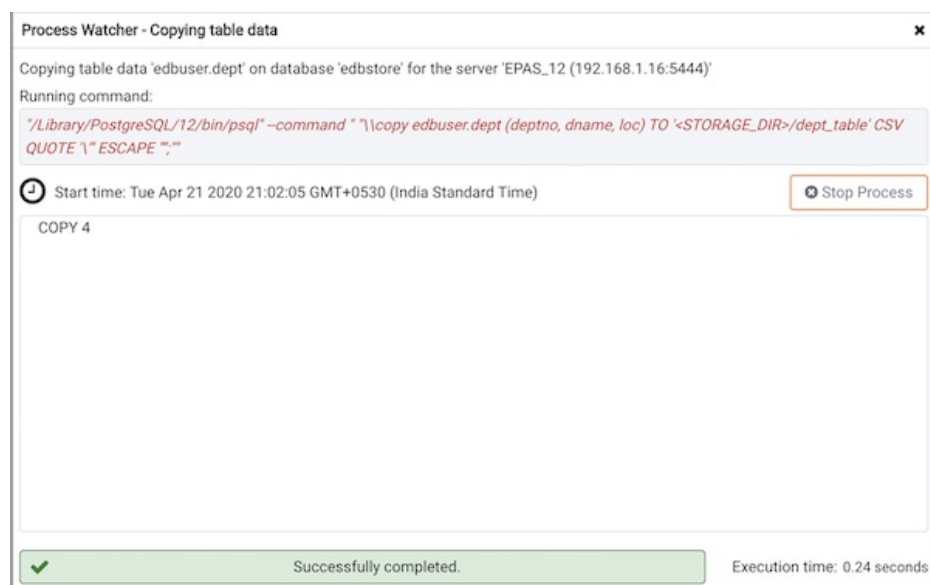
- Click inside the **Columns to export/import** field to deselect one or more columns from the drop-down listbox. To delete a selection, click the **x** to the left of the column name. Click an empty spot inside the field to access the drop-down list.
- Use the **NULL Strings** field to specify a string that will represent a null value within the source or target file.
- If enabled, click inside the **Not null columns** field to select one or more columns that will not be checked for a NULL value. To delete a column, click the **x** to the left of the column name.

After completing the **Import/Export data** dialog, click the **OK** button to perform the import or export. PEM will inform you when the background process completes:




Use the **Stop Process** button to stop the Import/Export process.

Use the [Click here for details](#) link on the notification to open the **Process Watcher** and review detailed information about the execution of the command that performed the import or export:

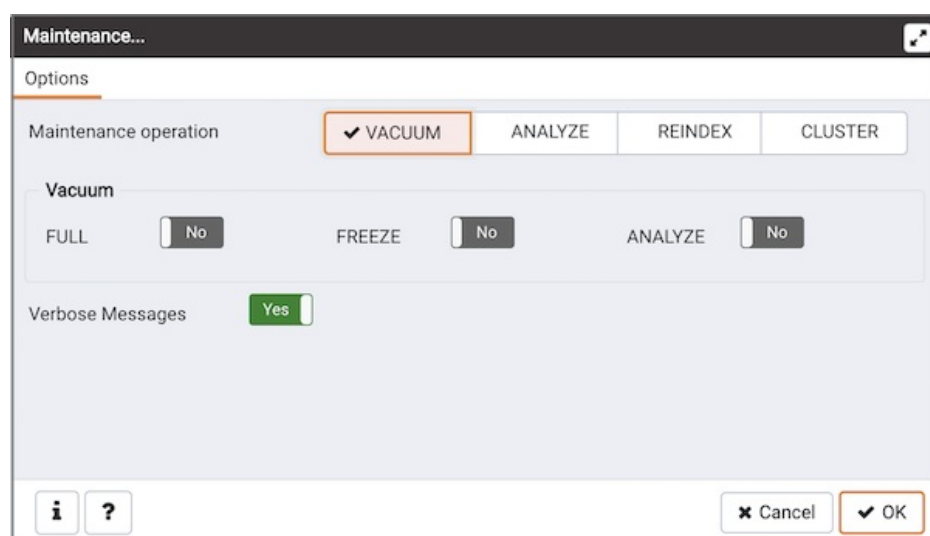


#### Note



- You can click on the  icon in the process watcher window to open the file location in the Storage Manager. You can use the [Storage Manager](#) to download the backup file on the client machine .
- If **ENABLE\_DATA\_ACCESS\_TOOLS** configuration option is set to False then Import/Export option is disabled.

## 34.5.4 Maintain a database object



This tool allows to maintain the database in total, or only a selected table, or a selected index.

Maintenance comes in three flavors.

### VACUUM

VACUUM will scan the database or table for rows, that are not in use any more. If a row is updated or deleted, the previous content isn't replaced, but rather marked invalid. The new data is inserted freshly into the database. You need to perform a garbage collection regularly, to insure that your database doesn't contain too much unused data, wasting disk space and ultimately degrading performance.

Please press the Help button to see the PostgreSQL help about the VACUUM command to learn more about the options.

The output of the database server is displayed in the messages page as they arrive. If Verbose is selected, the server will send very detailed info about what it did.

While this tool is very handy for ad-hoc maintenance purposes, you are encouraged to install an automatic job, that performs a VACUUM job regularly to keep your database in a neat state.

### ANALYZE

ANALYZE investigates statistical values about the selected database or table. This enables the query optimizer to select the fastest query plan, to give optimal performance. Every time your data is changing radically, you should perform this task. It can be included in a VACUUM run, using the appropriate option.

### REINDEX

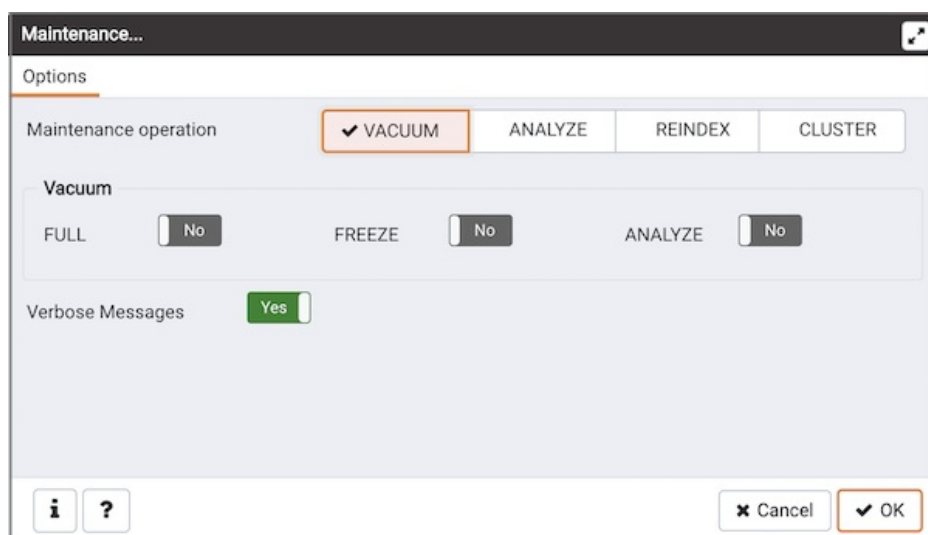
REINDEX rebuilds the indexes in case these have degenerated caused by unusual data patterns inserted. This can happen for example if you insert many rows with increasing index values, and delete low index values.

The RECREATE option doesn't call the REINDEX SQL command internally, instead it drops the existing table and recreates it according to the current index

definition. This doesn't lock the table exclusively, as REINDEX does, but will lock write access only.

### 34.5.4.1 Maintenance Dialog

Use the **Maintenance** dialog to VACUUM, ANALYZE, REINDEX or CLUSTER a database or selected database objects.



While this utility is useful for ad-hoc maintenance purposes, you are encouraged to perform automatic VACUUM jobs on a regular schedule.

Select a button next to **Maintenance operation** to specify the type of maintenance:

- Click **VACUUM** to scan the selected database or table to reclaim storage used by dead tuples.
  - Move the **FULL** switch to the **Yes** position to compact tables by writing a completely new version of the table file without dead space. The default is **No**.
  - Move the **FREEZE** switch to the **Yes** position to freeze data in a table when it will have no further updates. The default is **No**.
  - Move the **ANALYZE** switch to the **Yes** position to issue ANALYZE commands whenever the content of a table has changed sufficiently. The default is **No**.
- Click **ANALYZE** to update the stored statistics used by the query planner. This enables the query optimizer to select the fastest query plan for optimal performance.
- Click **REINDEX** to rebuild any index in case it has degenerated due to the insertion of unusual data patterns. This happens, for example, if you insert rows with increasing index values, and delete low index values.
- Click **CLUSTER** to instruct PostgreSQL to cluster the selected table.

To exclude status messages from the process output, move the **Verbose Messages** switch to the **No** position; by default, status messages are included.

When you've completed the dialog, click **OK** to start the background process; to exit the dialog without performing maintenance operations, click **Cancel**.

PEM will inform you when the background process completes:



Maintenance

Maintenance (Vacuum)  
Wed Mar 06 2019 10:36:31 GMT+0530 (India Standard Time)  
2.99 seconds
More details...
Stop Process

Successfully completed.

Use the **Stop Process** button to stop the Maintenance process.

Use the [Click here for details](#) link on the notification to open the **Process Watcher** and review detailed information about the execution of the command that performed the import or export:

Process Watcher - Maintenance

VACUUM (FULL, FREEZE, VERBOSE)  
Running Query:  
VACUUM FULL FREEZE VERBOSE ANALYZE;

Start time: Wed Mar 06 2019 10:47:57 GMT+0530 (India Standard Time)
Stop Process

INFO: "index\_size": scanned 9 of 9 pages, containing 791 live rows and 0 dead rows; 791 rows in sample, 791 estimated total rows  
INFO: vacuuming "pemhistory.os\_info"  
INFO: "os\_info": found 0 removable, 16 nonremovable row versions in 1 pages  
DETAIL: 0 dead row versions cannot be removed yet.  
CPU: user: 0.00 s, system: 0.00 s, elapsed: 0.00 s.  
INFO: analyzing "pemhistory.os\_info"  
INFO: "os\_info": scanned 1 of 1 pages, containing 16 live rows and 0 dead rows; 16 rows in sample, 16 estimated total rows  
INFO: vacuuming "pem.probe\_config\_table"  
INFO: "probe\_config\_table": found 0 removable, 0 nonremovable row versions in 0 pages  
DETAIL: 0 dead row versions cannot be removed yet.  
CPU: user: 0.00 s, system: 0.00 s, elapsed: 0.00 s.  
INFO: analyzing "pem.probe\_config\_table"  
INFO: "probe\_config\_table": scanned 0 of 0 pages, containing 0 live rows and 0 dead rows; 0 rows in sample, 0 estimated total rows

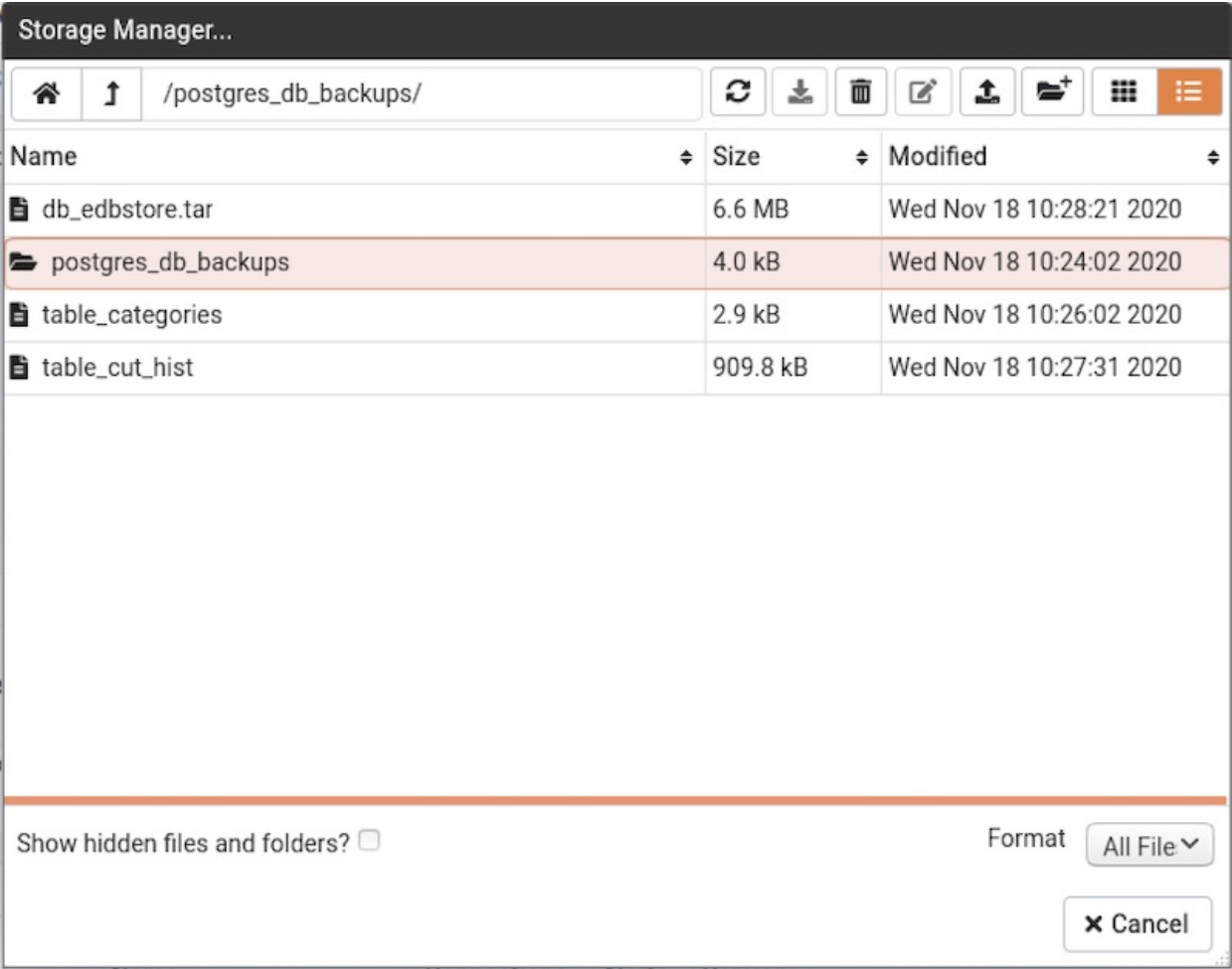
Successfully completed.
Execution time: 3.06 seconds

## 34.5.5 Storage Manager

*Storage Manager* is a feature that helps you manage your systems storage device. You can use *Storage Manager* to:

- Download, upload, or manage operating system files.
- Download *backup* or *export* files (custom, tar and plain text format) on a client machine.
- Download *export* dump files of tables.

You can access *Storage Manager* from the *Tools* Menu.



Use icons on the top of the *Storage Manager* window to manage storage:

Use the **Home** icon  to return to the home directory.

Use the **Up Arrow** icon  to return to the previous directory.

Use the **Refresh** icon  to display the most-recent files available.

Select the **Download** icon  to download the selected file.

Select the **Delete** icon  to delete the selected file or folder.

Select the **Edit** icon  to rename a file or folder.

Use the **Upload** icon  to upload a file.

Use the **New Folder** icon  to add a new folder.

Use the **Grid View** icon  to display all the files and folders in a grid view.

Use the **Table View** icon  to display all the files and folders in a list view.

Click on the check box next to *Show hidden files and folders* at the bottom of the window to view hidden files and folders.

Use the *Format* drop down list to select the format of the files to be displayed; choose from *sql*, *csv*, or *All Files*.

You can also download backup files through *Storage Manager* at the successful completion of the backups taken through [Backup Dialog](#), [Backup Global Dialog](#), or [Backup Server Dialog](#).

At the successful completion of a backup, click on the icon to open the current backup file in *Storage Manager* on the *process watcher* window.

Process Watcher - Backing up an object on the server

Backing up an object on the server 'PGSQL\_12 (localhost:5432)' from database 'edbstore'...

Running command:  

```
/usr/lib/postgresql/12/bin/pg_dump -file "/root/.pem/storage/pga_user_10/table_categories" -host "localhost" -port "5432" -username "postgres" --no-password --verbose --format=c -blobs -table "edbuser.categories" "edbstore"
```

Start time: Wed Nov 18 2020 10:26:02 GMT+0530 (India Standard Time)

Stop Process

```
pg_dump: reading policies for table "edbuser.categories"
pg_dump: reading row security enabled for table "edbuser.categories_category_seq"
pg_dump: reading policies for table "edbuser.categories_category_seq"
pg_dump: reading publications
pg_dump: reading publication membership
pg_dump: reading publication membership for table "edbuser.categories"
pg_dump: reading subscriptions
pg_dump: reading large objects
pg_dump: reading dependency data
pg_dump: saving encoding = UTF8
pg_dump: saving standard_conforming_strings = on
pg_dump: saving search_path =
pg_dump: saving database definition
pg_dump: dumping contents of table "edbuser.categories"
```

✓

Successfully completed.

Execution time: 0.14 seconds

### 34.5.6 Backup Dialog

PEM uses the *pg\_dump* utility to provide an easy way to create a backup in a plain-text or archived format. You can then use a client application (like *psql* or the *Query tool* ) to restore a plain-text backup file, or use the Postgres *pg\_restore* utility to restore an archived backup. The *pg\_dump* utility must have read access to all database objects that you want to back up.

You can backup a single table, a schema, or a complete database. Select the name of the backup source in the **Browser** tree control, right click to open the context menu, and select **Backup...** to open the **Backup** dialog. The name of the object selected will appear in the dialog title bar.

**Backup (Database: edbstore)**

General    Dump options

Filename: edbstore\_db

Format: Custom

Compression ratio:

Encoding: Select from the list

Number of jobs:

Role name: Select an item...

Buttons: [i] [?] [x] Cancel [Backup]

Use the fields in the **General** tab to specify parameters for the backup:

- Enter the name of the backup file in the **Filename** field. Optionally, select the **Browser** icon (...) to the right to navigate into a directory and select a file that will contain the archive.
- Use the drop-down listbox in the **Format** field to select the format that is best suited for your application. Each format has advantages and disadvantages:
  - Select **Custom** to create a custom archive file that you can use with **pg\_restore** to create a copy of a database. Custom archive file formats must be restored with **pg\_restore**. This format offers the opportunity to select which database objects to restore from the backup file. **Custom** archive format is recommended for medium to large databases as it is compressed by default.
  - Select **Tar** to generate a tar archive file that you can restore with **pg\_restore**. The tar format does not support compression.
  - Select **Plain** to create a plain-text script file. A plain-text script file contains SQL statements and commands that you can execute at the **psql** command line to recreate the database objects and load the table data. A plain-text backup file can be edited in a text editor, if desired, before using the **psql** program to restore database objects. **Plain** format is normally recommended for smaller databases; script dumps are not recommended for blobs. The SQL commands within the script will reconstruct the database to the last saved state of the database. A plain-text script can be used to reconstruct the database on another machine, or (with modifications) on other architectures.
  - Select **Directory** to generate a directory-format archive suitable for use with **pg\_restore**. This file format creates a directory with one file for each table and blob being dumped, plus a **Table of Contents** file describing the dumped objects in a machine-readable format that **pg\_restore** can read. This format is compressed by default.
- Use the **Compression Ratio** field to select a compression level for the backup. Specify a value of zero to mean use no compression; specify a maximum compression value of 9. Please note that tar archives do not support compression.
- Use the **Encoding** drop-down listbox to select the character encoding method that should be used for the archive.
- Use the **Number of Jobs** field (when applicable) to specify the number of tables that will be dumped simultaneously in a parallel backup.
- Use the dropdown listbox next to **Role name** to specify the role that owns the backup.

Click the **Dump options** tab to continue. Use the box fields in the **Dump options** tab to provide options for **pg\_dump**.

**Backup (Database: edbstore)**

General Dump options

**Sections**

|           |                             |      |                             |
|-----------|-----------------------------|------|-----------------------------|
| Pre-data  | <input type="checkbox"/> No | Data | <input type="checkbox"/> No |
| Post-data | <input type="checkbox"/> No |      |                             |

**i ?** **Cancel** **Backup**

- Move switches in the **Sections** field box to select a portion of the object that will be backed up.
  - Move the switch next to **Pre-data** to the **Yes** position to include all data definition items not included in the data or post-data item lists.
  - Move the switch next to **Data** to the **Yes** position to backup actual table data, large-object contents, and sequence values.
  - Move the switch next to **Post-data** to the **Yes** position to include definitions of indexes, triggers, rules, and constraints other than validated check constraints.

**Backup (Database: edbstore)**

General Dump options

**Type of objects**

|           |   |             |                             |
|-----------|---|-------------|-----------------------------|
| Only data | <input type="checkbox"/> No             | Only schema | <input type="checkbox"/> No |
| Blobs     | <input checked="" type="checkbox"/> Yes |             |                             |

**i ?** **Cancel** **Backup**

- Move switches in the **Type of objects** field box to specify details about the type of objects that will be backed up.
  - Move the switch next to **Only data** to the **Yes** position to limit the back up to data.
  - Move the switch next to **Only schema** to limit the back up to schema-level database objects.
  - Move the switch next to **Blobs** to the **No** position to exclude large objects in the backup.

**Backup (Database: edbstore)**

General Dump options

**Do not save**

|            |                             |                     |                             |
|------------|-----------------------------|---------------------|-----------------------------|
| Owner      | <input type="checkbox"/> No | Privilege           | <input type="checkbox"/> No |
| Tablespace | <input type="checkbox"/> No | Unlogged table data | <input type="checkbox"/> No |
| Comments   | <input type="checkbox"/> No |                     |                             |

**i ?** **Cancel** **Backup**

- Move switches in the **Do not save** field box to select the objects that will not be included in the backup.
  - Move the switch next to **Owner** to the **Yes** position to exclude commands that set object ownership.
  - Move the switch next to **Privilege** to the **Yes** position to exclude commands that create access privileges.
  - Move the switch next to **Tablespace** to the **Yes** position to exclude tablespaces.
  - Move the switch next to **Unlogged table data** to the **Yes** position to exclude the contents of unlogged tables.
  - Move the switch next to **Comments** to the **Yes** position to exclude commands that set the comments. **Note:** This option is visible only for database server greater than or equal to 11.

**Backup (Database: edbstore)**

General Dump options

**Queries**

|                                   |                             |                                 |                             |
|-----------------------------------|-----------------------------|---------------------------------|-----------------------------|
| Use Column Inserts                | <input type="checkbox"/> No | Use Insert Commands             | <input type="checkbox"/> No |
| Include CREATE DATABASE statement | <input type="checkbox"/> No | Include DROP DATABASE statement | <input type="checkbox"/> No |
| Load Via Partition Root           | <input type="checkbox"/> No |                                 |                             |

**i ?** **Cancel** **Backup**

- Move switches in the **Queries** field box to specify the type of statements that should be included in the backup.
  - Move the switch next to **Use Column Inserts** to the **Yes** position to dump the data in the form of INSERT statements and include explicit column names. Please note: this may make restoration from backup slow.
  - Move the switch next to **Use Insert commands** to the **Yes** position to dump the data in the form of INSERT statements rather than using a COPY command. Please note: this may make restoration from backup slow.
  - Move the switch next to **Include CREATE DATABASE statement** to the **Yes** position to include a command in the backup that creates a new database when restoring the backup.
  - Move the switch next to **Include DROP DATABASE statement** to the **Yes** position to include a command in the backup that will drop any existing database object with the same name before recreating the object during a backup.
  - Move the switch next to **Load Via Partition Root** to the **Yes** position, so when dumping a COPY or INSERT statement for a partitioned table, target the root of the partitioning hierarchy which contains it rather than the partition itself. **Note:** This option is visible only

for database server greater than or equal to 11.

**Backup (Database: edbstore)**

General Dump options

**Disable**

Trigger ☐ No \$ quoting ☐ No

? × Cancel Backup

- Move switches in the **Disable** field box to specify the type of statements that should be excluded from the backup.
  - Move the switch next to **Trigger** (active when creating a data-only backup) to the **Yes** position to include commands that will disable triggers on the target table while the data is being loaded.
  - Move the switch next to **\$ quoting** to the **Yes** position to enable dollar quoting within function bodies; if disabled, the function body will be quoted using SQL standard string syntax.

**Backup (Database: edbstore)**

General Dump options

**Miscellaneous**

With OID(s) ☐ No Verbose messages ☒ Yes

Force double quote on identifiers ☐ No Use SET SESSION AUTHORIZATION ☐ No

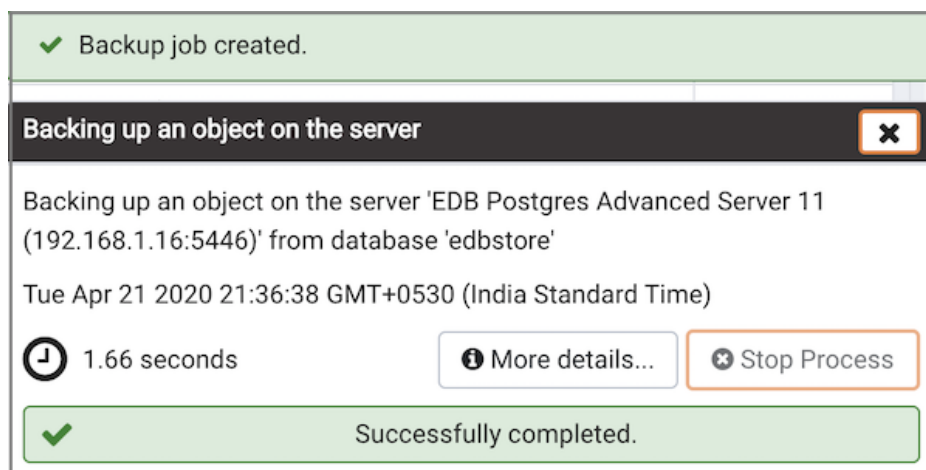
? × Cancel Backup

- Move switches in the **Miscellaneous** field box to specify miscellaneous backup options.
  - Move the switch next to **With OIDs** to the **Yes** position to include object identifiers as part of the table data for each table.
  - Move the switch next to **Verbose messages** to the **No** position to instruct `pg_dump` to exclude verbose messages.
  - Move the switch next to **Force double quotes on identifiers** to the **Yes** position to force the quoting of all identifiers.
  - Move the switch next to **Use SET SESSION AUTHORIZATION** to the **Yes** position to include a statement that will use a SET SESSION AUTHORIZATION command to determine object ownership (instead of an ALTER OWNER command).

When you've specified the details that will be incorporated into the `pg_dump` command:

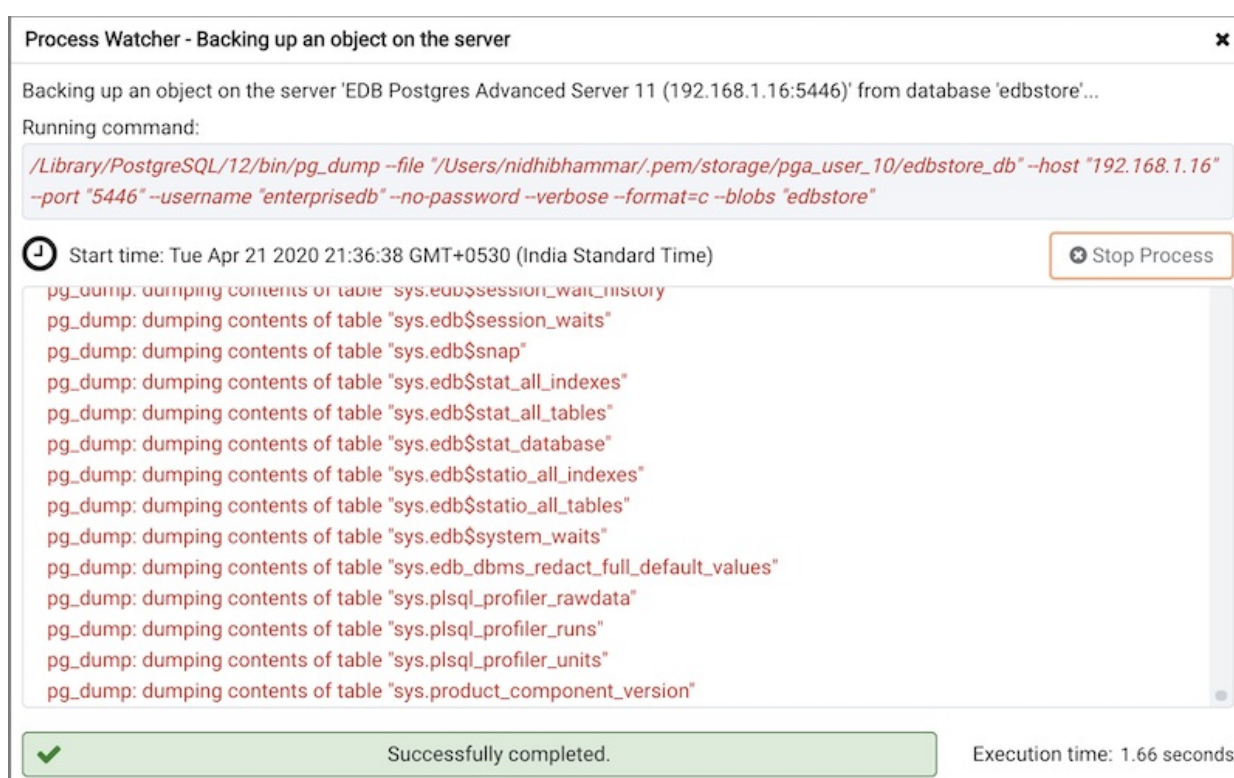
- Click the **Backup** button to build and execute a command that builds a backup based on your selections on the **Backup** dialog.
- Click the **Cancel** button to exit without saving work.





Use the **Stop Process** button to stop the Backup process.


If the backup is successful, a popup window will confirm success. Click *More details* on the popup window to launch the *Process Watcher*. The *Process Watcher* logs all the activity associated with the backup and provides additional information for troubleshooting.



If the backup is unsuccessful, you can review the error messages returned by the backup command on the [Process Watcher](#).

#### Note



- You can click on the  icon in the process watcher window to open the file location in the Storage Manager. You can use the [Storage Manager](#) to download the backup file on the client machine.
- If `ENABLE_DATA_ACCESS_TOOLS` configuration option is set to False then backup option is disabled.

## 34.5.7 Backup Globals Dialog



Use the **Backup Globals** dialog to create a plain-text script that recreates all of the database objects within a cluster, and the global objects that are shared by those databases. Global objects include tablespaces, roles, and object properties. You can use the PEM **Query tool** to play back a plain-text script, and recreate the objects in the backup.

Use the fields in the **General** tab to specify the following:

- Enter the name of the backup file in the **Filename** field. Optionally, select the **Browser** icon (ellipsis) to the right to navigate into a directory and select a file that will contain the archive.
- Use the drop-down listbox next to **Role name** to specify a role with connection privileges on the selected server. The role will be used for authentication during the backup.

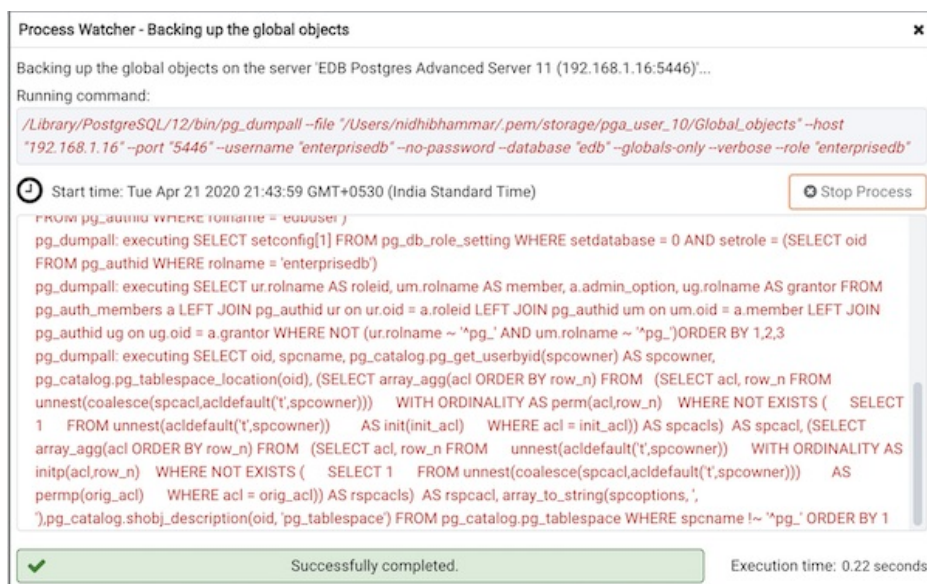
Move switches in the **Miscellaneous** field box to specify the type of statements that should be included in the backup.

- Move the **Verbose messages** switch to the **No** position to exclude status messages from the backup. The default is **Yes**.
- Move the **Force double quote on identifiers** switch to the **Yes** position to name identifiers without changing case. The default is **No**.

Click the **Backup** button to build and execute a command based on your selections; click the **Cancel** button to exit without saving work.

Use the **Stop Process** button to stop the Backup process.


If the backup is successful, a popup window will confirm success. Click [Click here for details](#) on the popup window to launch the **Process Watcher**. The **Process Watcher** logs all the activity associated with the backup and provides additional information for troubleshooting.



If the backup is unsuccessful, review the error message returned by the **Process Watcher** to resolve any issue.

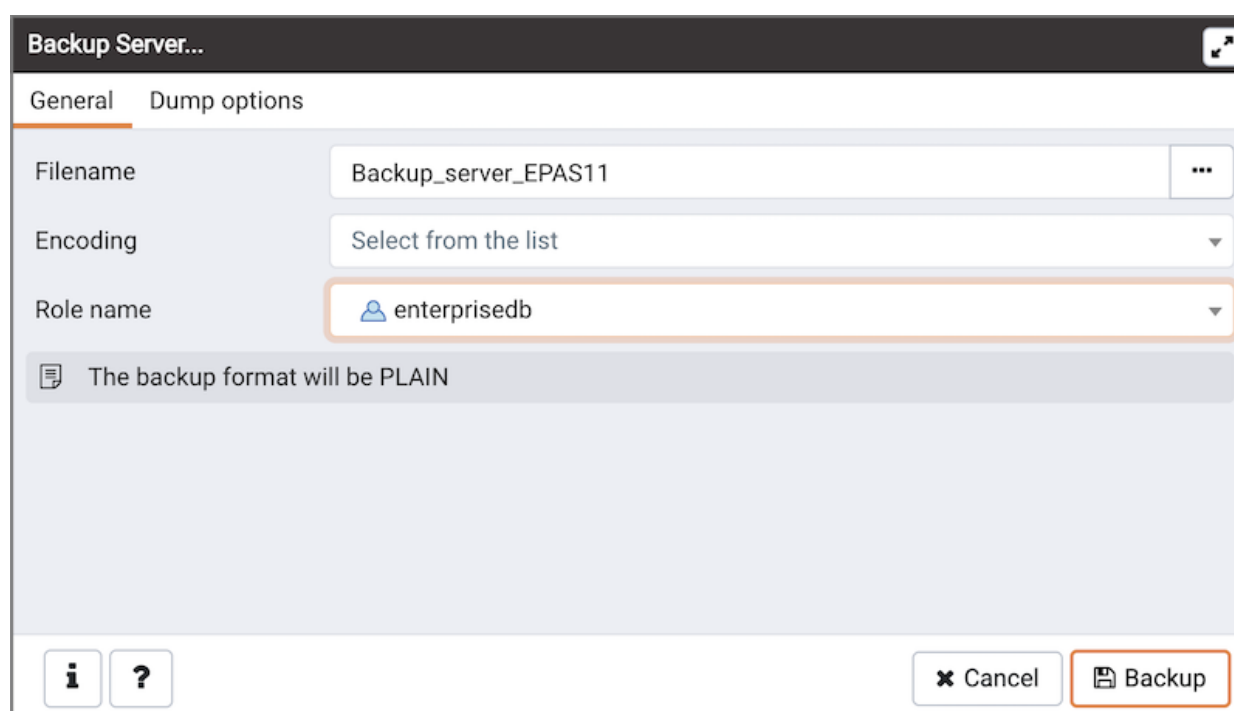
#### Note



- You can click on the  icon in the process watcher window to open the file location in the Storage Manager. You can use the [Storage Manager](#) to download the backup file on the client machine .
- If **ENABLE\_DATA\_ACCESS\_TOOLS** configuration option is set to False then backup option is disabled.

## 34.5.8 Backup Server Dialog

Use the **Backup Server** dialog to create a plain-text script that will recreate the selected server. You can use the PEM **Query tool** to play back a plain-text script, and recreate the server.



Use the fields in the **General** tab to specify the following:

- Enter the name of the backup file in the **Filename** field. Optionally, select the **Browser** icon (ellipsis) to the right to navigate into a directory and select a file that will contain the archive.
- Use the **Encoding** drop-down listbox to select the character encoding method that should be used for the archive. **Note:** This option is visible only for database server greater than or equal to 11.
- Use the drop-down listbox next to **Role name** to specify a role with connection privileges on the selected server. The role will be used for authentication during the backup.

**Backup Server...**

General **Dump options**

**Type of objects**

Only data ☐ No      Only schema ☐ No

**i** **?** **Cancel** **Backup**

- Move switches in the **Type of objects** field box to specify details about the type of objects that will be backed up.
  - Move the switch next to **Only data** to the **Yes** position to limit the back up to data.
  - Move the switch next to **Only schema** to limit the back up to schema-level database objects.

**Backup Server...**

General **Dump options**

**Do not save**

Owner ☐ No      Privilege ☐ No

Tablespace ☐ No      Unlogged table data ☐ No

Comments ☐ No

**i** **?** **Cancel** **Backup**

- Move switches in the **Do not save** field box to select the objects that will not be included in the backup.
  - Move the switch next to **Owner** to the **Yes** position to exclude commands that set object ownership.
  - Move the switch next to **Privilege** to the **Yes** position to exclude commands that create access privileges.
  - Move the switch next to **Tablespace** to the **Yes** position to exclude tablespaces.
  - Move the switch next to **Unlogged table data** to the **Yes** position to exclude the contents of unlogged tables.
  - Move the switch next to **Comments** to the **Yes** position to exclude commands that set the comments. **Note:** This option is visible only for database server greater than or equal to 11.

**Backup Server...**

General Dump options

**Queries**

Use Column Inserts ☐ No

Use Insert Commands ☐ No

Include DROP DATABASE statement ☐ No

**i ?** **Cancel Backup**

- Move switches in the **Queries** field box to specify the type of statements that should be included in the backup.

- Move the switch next to **Use Column Inserts** to the **Yes** position to dump the data in the form of INSERT statements and include explicit column names. Please note: this may make restoration from backup slow.
- Move the switch next to **Use Insert commands** to the **Yes** position to dump the data in the form of INSERT statements rather than using a COPY command. Please note: this may make restoration from backup slow.
- Move the switch next to **Include DROP DATABASE statement** to the **Yes** position to include a command in the backup that will drop any existing database object with the same name before recreating the object during a backup.

**Backup Server...**

General Dump options

**Disable**

Trigger ☐ No

\$ quoting ☐ No

**i ?** **Cancel Backup**

- Move switches in the **Disable** field box to specify the type of statements that should be excluded from the backup.

- Move the switch next to **Trigger** (active when creating a data-only backup) to the **Yes** position to include commands that will disable triggers on the target table while the data is being loaded.
- Move the switch next to **\$ quoting** to the **Yes** position to enable dollar quoting within function bodies; if disabled, the function body will be quoted using SQL standard string syntax.

**Backup Server...**

General Dump options

**Miscellaneous**

With OID(s) ☐ No ☒ Yes

Force double quote on identifiers ☐ No ☒ Yes

Verbose messages ☒ Yes ☐ No

Use SET SESSION AUTHORIZATION ☐ No ☒ Yes

**i ?** **Cancel** **Backup**

- Move switches in the **Miscellaneous** field box to specify miscellaneous backup options.

- Move the switch next to **With OIDs** to the **Yes** position to include object identifiers as part of the table data for each table.
- Move the switch next to **Verbose messages** to the **No** position to instruct `pg_dump` to exclude verbose messages.
- Move the switch next to **Force double quotes on identifiers** to the **Yes** position to force the quoting of all identifiers.
- Move the switch next to **Use SET SESSION AUTHORIZATION** to the **Yes** position to include a statement that will use a SET SESSION AUTHORIZATION command to determine object ownership (instead of an ALTER OWNER command).

Click the **Backup** button to build and execute a command based on your selections; click the **Cancel** button to exit without saving work.

**Backing up the server**

Backing up the server 'EDB Postgres Advanced Server 11 (192.168.1.16:5446)'

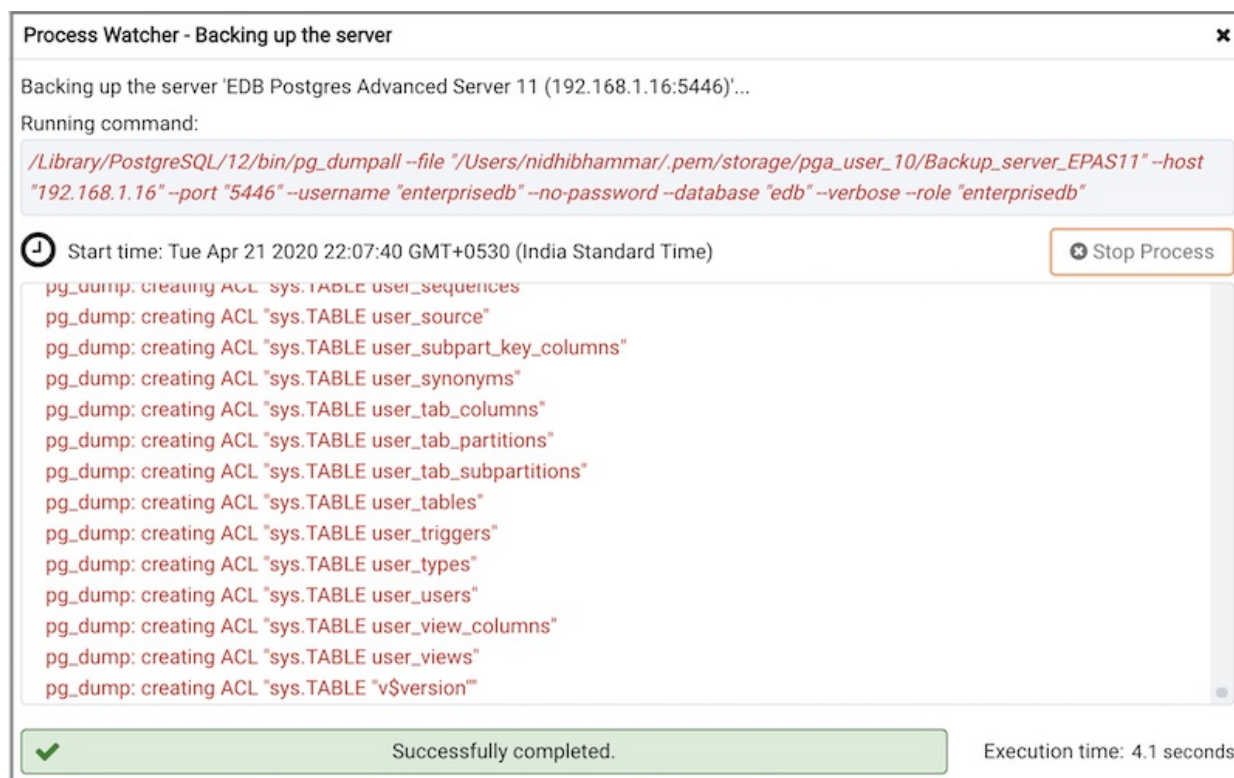
Tue Apr 21 2020 22:07:40 GMT+0530 (India Standard Time)

⌚ 4.1 seconds **More details...** **Stop Process**

✓ **Successfully completed.**

Use the **Stop Process** button to stop the Backup process.


If the backup is successful, a popup window will confirm success. Click [Click here for details](#) on the popup window to launch the **Process Watcher**. The **Process Watcher** logs all the activity associated with the backup and provides additional information for troubleshooting.



If the backup is unsuccessful, review the error message returned by the [Process Watcher](#) to resolve any issue.

#### Note



- You can click on the  icon in the process watcher window to open the file location in the Storage Manager. You can use the [Storage Manager](#) to download the backup file on the client machine .
- If `ENABLE_DATA_ACCESS_TOOLS` configuration option is set to False then backup option is disabled.

## 34.5.9 Restore Dialog

The [Restore](#) dialog provides an easy way to use a Custom, tar, or Directory format backup taken with the PEM [Backup](#) dialog to recreate a database or database object. The [Backup](#) dialog invokes options of the `pg_dump` client utility; the [Restore](#) dialog invokes options of the `pg_restore` client utility.

You can use the [Query tool](#) to play back the script created during a plain-text backup made with the [Backup](#) dialog. For more information about backing up or restoring, please refer to the documentation for [pg\\_dump](#) or [pg\\_restore](#).

Restore (Database: db01)

General

Restore options

Format

Custom or tar

Filename

/edbstore\_db\_file

...

Number of jobs

Role name

Select an item...

i

?

Cancel

Restore

Use the fields on the **General** tab to specify general information about the restore process:

- Use the drop-down listbox in the **Format** field to select the format of your backup file.
  - Select **Custom or tar** to restore from a custom archive file to create a copy of the backed-up object.
  - Select **Directory** to restore from a compressed directory-format archive.
- Enter the complete path to the backup file in the **Filename** field. Optionally, select the **Browser** icon (ellipsis) to the right to navigate into a directory and select the file that contains the archive.
- Use the **Number of Jobs** field to specify if pg\_restore should use multiple (concurrent) jobs to process the restore. Each job uses a separate connection to the server.
- Use the drop-down listbox next to **Rolename** to specify the role that will be used to authenticate with the server during the restore process.

Click the **Restore options** tab to continue. Use the fields on the **Restore options** tab to specify options that correspond to **pg\_restore** options.

Restore (Database: edbstore)

General

Restore options

Sections

Pre-data

No

Data

No

Post-data

No

i

?

Cancel

Restore

- Use the switches in the **Sections** box to specify the content that will be restored:
  - Move the switch next to **Pre-data** to the **Yes** position to restore all data definition items not included in the data or post-data

item lists.

- Move the switch next to **Data** to the **Yes** position to restore actual table data, large-object contents, and sequence values.
- Move the switch next to **Post-data** to the **Yes** position to restore definitions of indexes, triggers, rules, and constraints (other than validated check constraints).

**Restore (Database: edbstore)**

General **Restore options**

**Type of objects**

Only data ☐ No      Only schema ☐ No

**i** **?** **Cancel** **Restore**

- Use the switches in the **Type of objects** box to specify the objects that will be restored:

- Move the switch next to **Only data** to the **Yes** position to limit the restoration to data.
- Move the switch next to **Only schema** to limit the restoration to schema-level database objects.

**Restore (Database: edbstore)**

General **Restore options**

**Do not save**

Owner ☐ No      Privilege ☐ No  
 Tablespace ☐ No      Comments ☐ No

**i** **?** **Cancel** **Restore**

- Use the switches in the **Do not save** box to specify which objects will not be restored:

- Move the switch next to **Owner** to the **Yes** position to exclude commands that set object ownership.
- Move the switch next to **Privilege** to the **Yes** position to exclude commands that create access privileges.
- Move the switch next to **Tablespace** to the **Yes** position to exclude tablespaces.
- Move the switch next to **Comments** to the **Yes** position to exclude commands that set the comments. **Note:** This option is visible only for database server greater than or equal to 11.



**Restore (Database: edbstore)**

General **Restore options**

**Queries**

Include CREATE DATABASE statement ☐ No

Clean before restore ☐ No

Single transaction ☐ No

**i ?** **Cancel** **Restore**

- Use the switches in the **Queries** box to specify the type of statements that should be included in the restore:

- Move the switch next to **Include CREATE DATABASE statement** to the **Yes** position to include a command that creates a new database before performing the restore.
- Move the switch next to **Clean before restore** to the **Yes** position to drop each existing database object (and data) before restoring.
- Move the switch next to **Single transaction** to the **Yes** position to execute the restore as a single transaction (that is, wrap the emitted commands in **BEGIN/COMMIT**). This ensures that either all the commands complete successfully, or no changes are applied. This option implies **--exit-on-error**.

**Restore (Database: edbstore)**

General **Restore options**

**Disable**

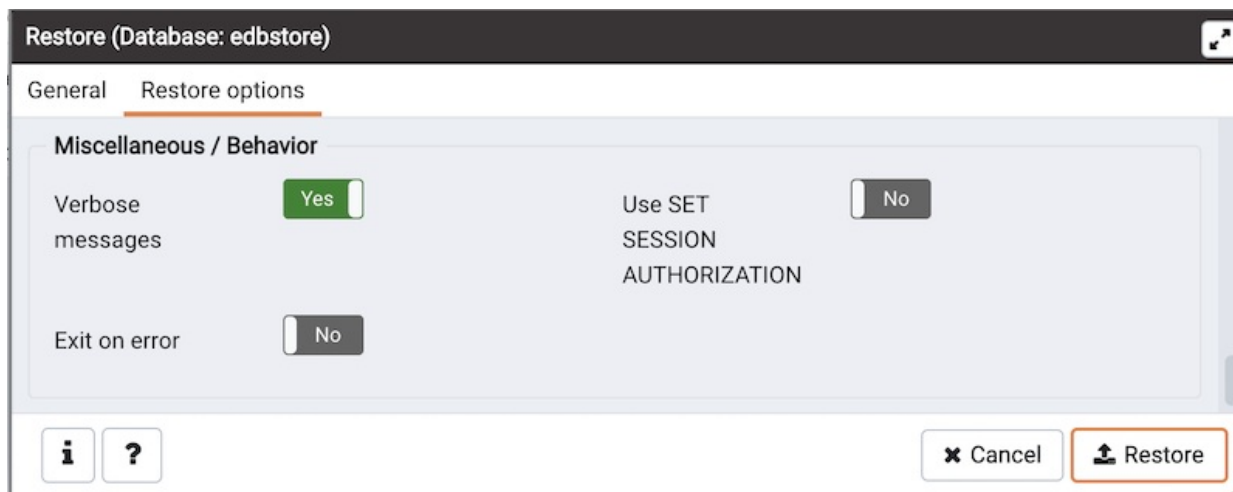
Trigger ☐ No

No data for Failed Tables ☐ No

**i ?** **Cancel** **Restore**

- Use the switches in the **Disable** box to specify the type of statements that should be excluded from the restore:

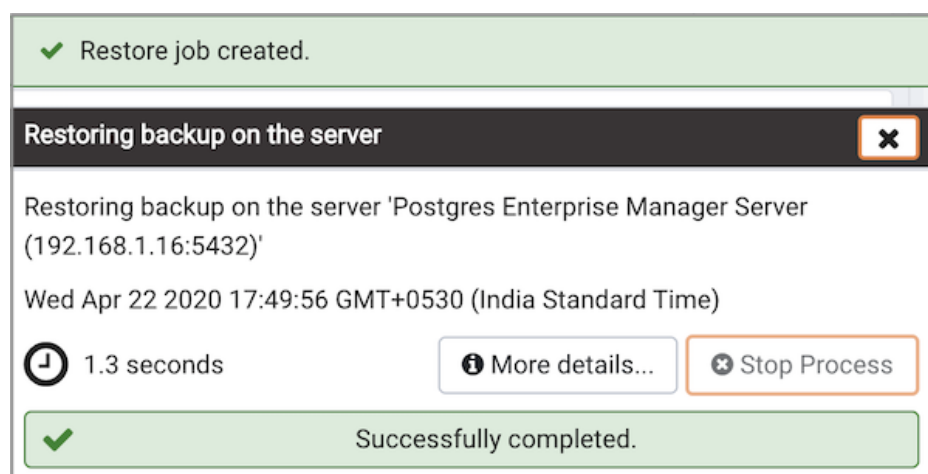
- Move the switch next to **Trigger** (active when creating a data-only restore) to the **Yes** position to include commands that will disable triggers on the target table while the data is being loaded.
- Move the switch next to **No data for Failed Tables** to the **Yes** position to ignore data that fails a trigger.



- Use the switches in the **Miscellaneous/Behavior** box to specify miscellaneous restore options:

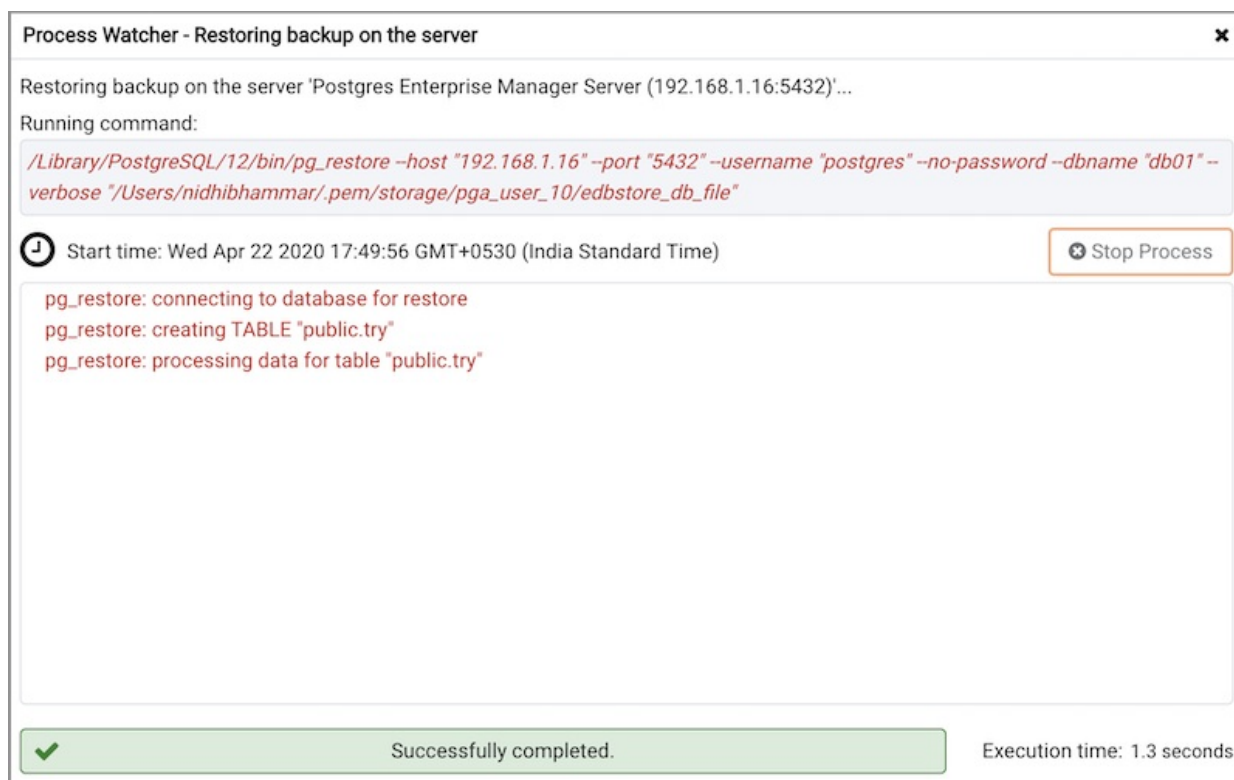
- Move the switch next to **Verbose messages** to the **No** position to instruct `pg_restore` to exclude verbose messages.
- Move the switch next to **Use SET SESSION AUTHORIZATION** to the **Yes** position to include a statement that will use a `SET SESSION AUTHORIZATION` command to determine object ownership (instead of an `ALTER OWNER` command).
- Move the switch next to **Exit on error** to the **Yes** position to instruct `pg_restore` to exit restore if there is an error in sending SQL commands. The default is to continue and to display a count of errors at the end of the restore.

When you've specified the details that will be incorporated into the `pg_restore` command, click the **Restore** button to start the process, or click the **Cancel** button to exit without saving your work. A popup will confirm if the restore is successful.



Use the **Stop Process** button to stop the Restore process.

Click [Click here for details](#) on the popup to launch the **Process Watcher**. The **Process Watcher** logs all the activity associated with the restore, and provides additional information for troubleshooting should the restore command encounter problems.



### 34.5.10 Managing Cluster Level Objects

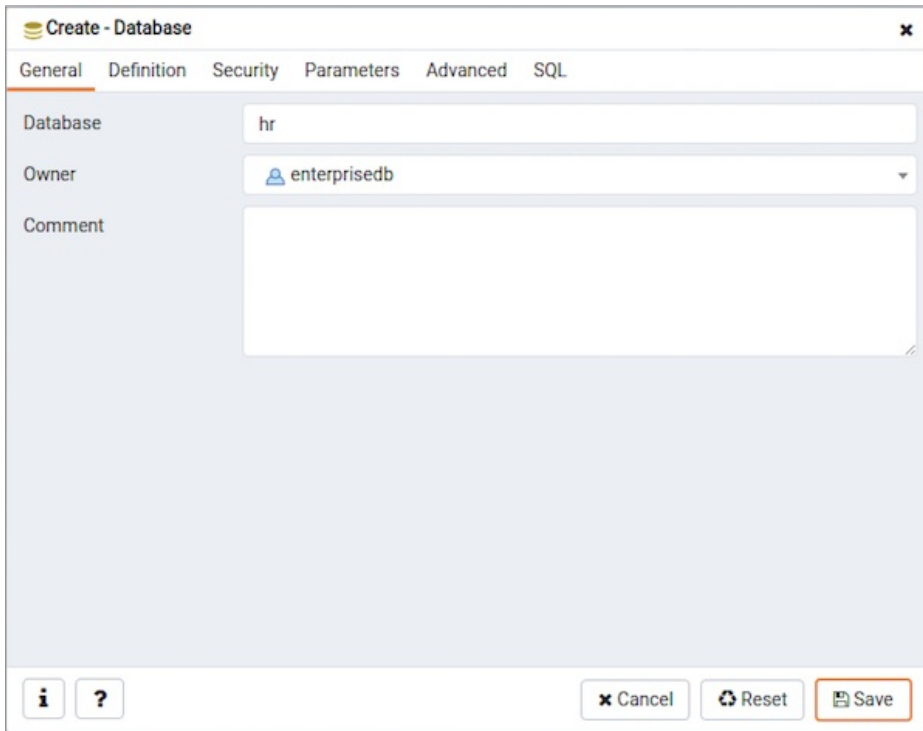
Some object definitions reside at the cluster level; PEM provides dialogs that allow you to create these objects, manage them, and control their relationships to each other. To access a dialog that allows you to create a database object, right-click on the object type in the Browser tree control, and select the **Create** option for that object. For example, to create a new database, right-click on the **Databases** node, and select **Create Database...**

Contents:

#### 34.5.10.1 Database Dialog

Use the **Database** dialog to define or modify a database. To create a database, you must be a database superuser or have the CREATE privilege.

The **Database** dialog organizes the development of a database through the following dialog tabs: **General**, **Definition**, **Security**, and **Parameters**. The **SQL** tab displays the SQL code generated by dialog selections.



The screenshot shows the 'Create - Database' dialog box with the following details:

- Title Bar:** Create - Database
- Tabs:** General (selected), Definition, Security, Parameters, Advanced, SQL
- Fields:**
  - Database:** Text input field containing 'hr'.
  - Owner:** Drop-down listbox showing 'enterisedb' with a user icon.
  - Comment:** Large text area for notes.
- Buttons:** Information (i), Help (?), Cancel, Reset, and Save (highlighted with an orange border).

Use the fields in the **General** tab to identify the database:

- Use the **Database** field to add a descriptive name for the database. The name will be displayed in the **Browser** tree control.
- Select the owner of the database from the drop-down listbox in the **Owner** field.
- Store notes about the database in the **Comment** field.

Click the **Definition** tab to continue.

The screenshot shows the 'Create - Database' dialog box with the 'Definition' tab selected. The dialog has five tabs: General, Definition, Security, Parameters, and SQL. The 'Definition' tab contains the following fields:

- Encoding:** A text field with 'UTF8' and a dropdown arrow.
- Template:** A dropdown menu with 'Select an item...'.
- Tablespace:** A dropdown menu with 'Select an item...'.
- Collation:** A dropdown menu with 'Select from the list'.
- Character type:** A dropdown menu with 'Select from the list'.
- Connection limit:** A text field with '-1'.

At the bottom of the dialog, there are four buttons: an information icon (i), a question mark icon (?), a 'Cancel' button, a 'Reset' button, and a 'Save' button (which is highlighted with an orange border).

Use the **Definition** tab to set properties for the database:

- Select a character set from the drop-down listbox in the **Encoding** field. The default is **UTF8**.
- Select a template from the drop-down listbox in the **Template** field. If you do not specify a template, the database will use template1.
- Select a tablespace from the drop-down listbox in the **Tablespace** field. The selected tablespace will be the default tablespace used to contain database objects.
- Select the collation order from the drop-down listbox in the **Collation** field.
- Select the character classification from the drop-down listbox in the **Character Type** field. This affects the categorization of characters, e.g. lower, upper and digit. The default, or a blank field, uses the character classification of the template database.
- Specify a connection limit in the **Connection Limit** field to configure the maximum number of connection requests. The default value (**-1**) allows unlimited connections to the database.

Click the **Security** tab to continue.

**Create - Database**

General Definition **Security** Parameters SQL

**Privileges** +

| Grantee   | Privileges  | Grantor  |
|-----------|---|----------|
| pem_admin | <input checked="" type="checkbox"/> ALL <input type="checkbox"/> WITH GRANT OPTION<br><input checked="" type="checkbox"/> CREATE <input type="checkbox"/> WITH GRANT OPTION<br><input checked="" type="checkbox"/> TEMPORARY <input type="checkbox"/> WITH GRANT OPTION<br><input checked="" type="checkbox"/> CONNECT <input type="checkbox"/> WITH GRANT OPTION | postgres |

**Security labels** +

| Provider | Security label |
|----------|----------------|
|----------|----------------|

Cancel
 Reset
 Save

Use the **Security** tab to assign privileges and define security labels.

Use the **Privileges** panel to assign privileges to a role. Click the **Add** icon (+) to set privileges for database objects:

- Select the name of the role from the drop-down listbox in the **Grantee** field.
- Click inside the **Privileges** field. Check the boxes to the left of one or more privileges to grant the selected privilege to the specified user.
- The current user, who is the default grantor for granting the privilege, is displayed in the **Grantor** field.

Click add to set additional privileges; to discard a privilege, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Use the **Security Labels** panel to define security labels applied to the database. Click the **Add** icon (+) to add each security label selection:

- Specify a security label provider in the **Provider** field. The named provider must be loaded and must consent to the proposed labeling operation.
- Specify a security label in the **Security Label** field. The meaning of a given label is at the discretion of the label provider. PostgreSQL places no restrictions on whether or how a label provider must interpret security labels; it merely provides a mechanism for storing them.

To discard a security label, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Click the **Parameters** tab to continue.

Create - Database

General

Definition

Security

Parameters

SQL

+

|  | Name                | Value | Role       |
|--|---------------------|-------|------------|
|  | statement_timeout ▾ | 30    | postgres ▾ |

i

?

✕ Cancel

↺ Reset

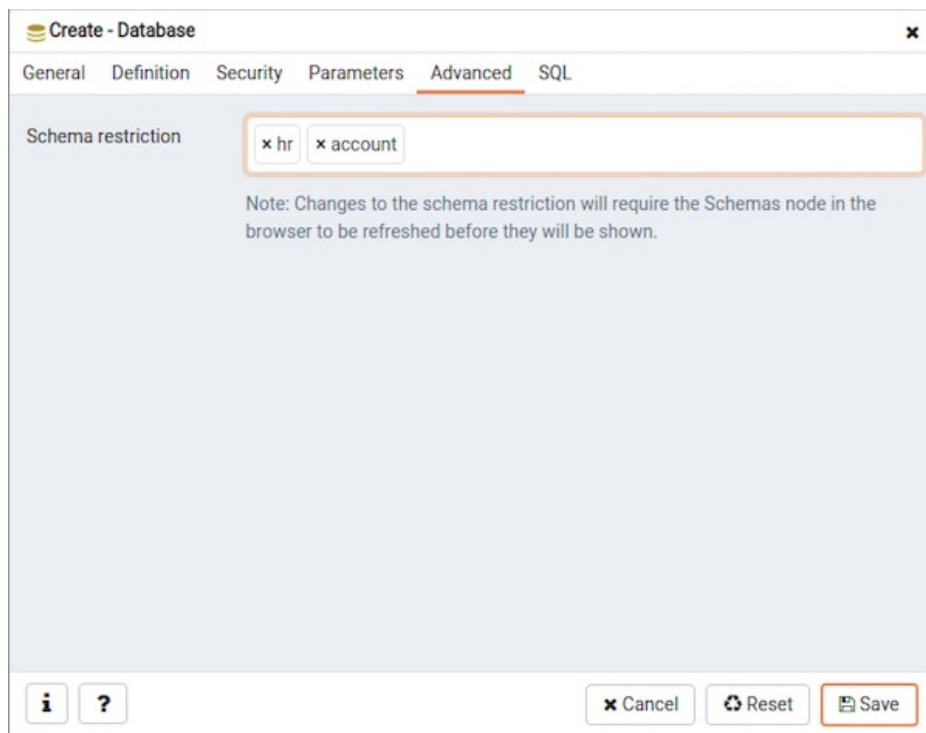
💾 Save

Use the **Parameters** tab to set parameters for the database. Click the **Add** icon (+) to add each parameter:

- Use the drop-down listbox in the **Name** field to select a parameter.
- Use the **Value** field to set a value for the parameter.
- Use the drop-down listbox next to **Role** to select a role to which the parameter setting specified will apply.

Follow these steps to add additional parameter value definitions; to discard a parameter, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Click the **Advanced** tab to continue.



Use the **Advanced** tab to set advanced parameters for the database.

- Use **Schema restriction** field to provide a SQL restriction that will be used against the `pg_namespace` table to limit the schemas that you see. For example, you might enter: `public` so that only `public` are shown in the browser tree. Separate entries with a comma or tab as you type.

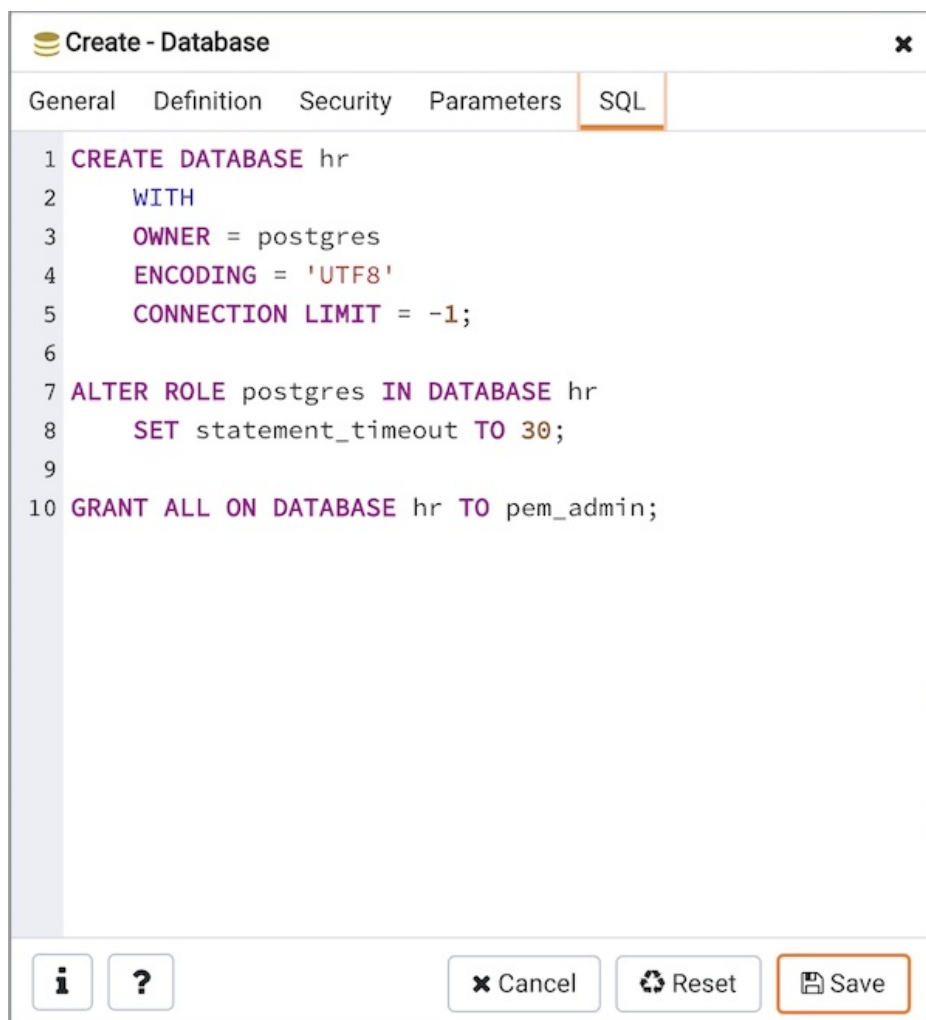
Click the **SQL** tab to continue.

Your entries in the **Database** dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

### Example

The following is an example of the sql command generated by user selections in the **Database** dialog:





The example creates a database named `hr` that is owned by `postgres`. It allows unlimited connections, and is available to all authenticated users.

- Click the **Info** button (i) to access online help. View context-sensitive help in the **Tabbed browser**, where a new tab displays the PostgreSQL core documentation.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

### 34.5.10.2 Move Objects Dialog

Use the **Move Objects** dialog to move database objects from one tablespace to another tablespace.

The **Move Objects** dialog organizes the movement of database objects with the **General** tab; the **SQL** tab displays the SQL code generated by dialog selections.

Use the fields in the **General** tab to identify the items that will be moved and the tablespace to which they will be moved:

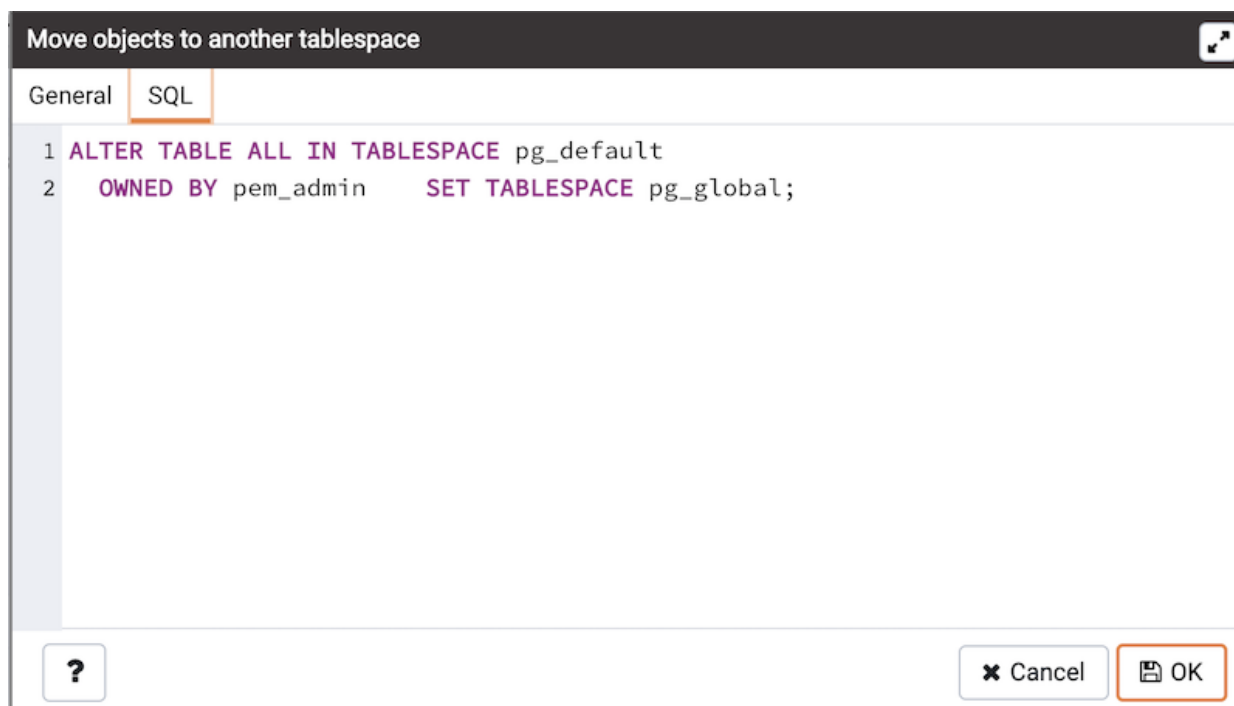
- Use the **New tablespace** drop-down listbox to select a pre-existing tablespace to which the object will be moved. (To create a tablespace, use the **Tablespace** dialog; access the dialog by right clicking **Tablespaces** in the **Browser** tree control and selecting **Create Tablespace...** from the context-menu.)
- Use the **Object type** drop-down listbox to select from the following:
  - Select **All** to move all tables, indexes, and materialized views from the current tablespace (currently selected in the **Browser** tree control) to the new tablespace.
  - Select **Tables** to move tables from the current tablespace to the new tablespace.
  - Select **Indexes** to move indexes from the current tablespace to the new tablespace.
  - Select **Materialized views** to move materialized views from the current tablespace to the new tablespace.
- Use the **Object owner** drop-down listbox to select the role that owns the objects selected in the **Object type** field. This field is optional.

Click the **SQL** tab to continue.

Your entries in the **Move Objects** dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit the **General** tab to modify the SQL command.

#### Example

The following is an example of the sql command generated by user selections in the **Move Objects** dialog:



The example shown demonstrates moving materialized views owned by Alice from tablespace `tblspace_01` to `tblspace_02`.

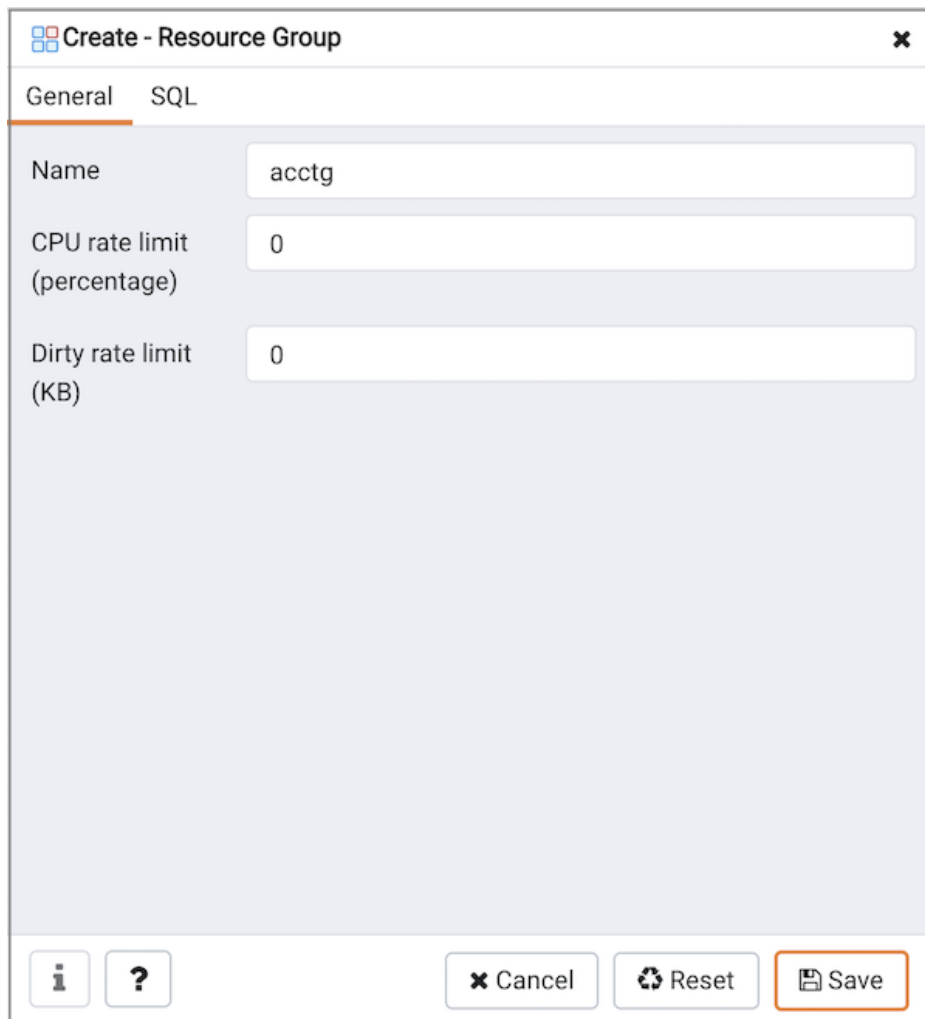
- Click the **Help** button (?) to access online help.
- Click the **OK** button to save work.
- Click the **Cancel** button to exit without saving work.

### 34.5.10.3 Resource Group Dialog

Use the **Resource Group** dialog to create a resource group and set values for its resources. A resource group is a named, global group on which various resource usage limits can be defined. The resource group is accessible from all databases in the cluster. To use the **Resource Group** dialog, you must have superuser privileges. Please note that resource groups are supported when connected to EDB Postgres Advanced Server; for more information about using resource groups, please see the EDB Postgres Advanced Server Guide, available at:

<http://www.enterprisedb.com/>

Fields used to create a resource group are located on the **General** tab. The **SQL** tab displays the SQL code generated by your selections on the **Resource Group** dialog.



The image shows a 'Create - Resource Group' dialog box with two tabs: 'General' and 'SQL'. The 'General' tab is active. It contains three input fields: 'Name' with the value 'acctg', 'CPU rate limit (percentage)' with the value '0', and 'Dirty rate limit (KB)' with the value '0'. At the bottom, there are buttons for 'Cancel', 'Reset', and 'Save' (which is highlighted with an orange border). There are also information and help icons on the left.

Use the fields on the **General** tab to specify resource group attributes:

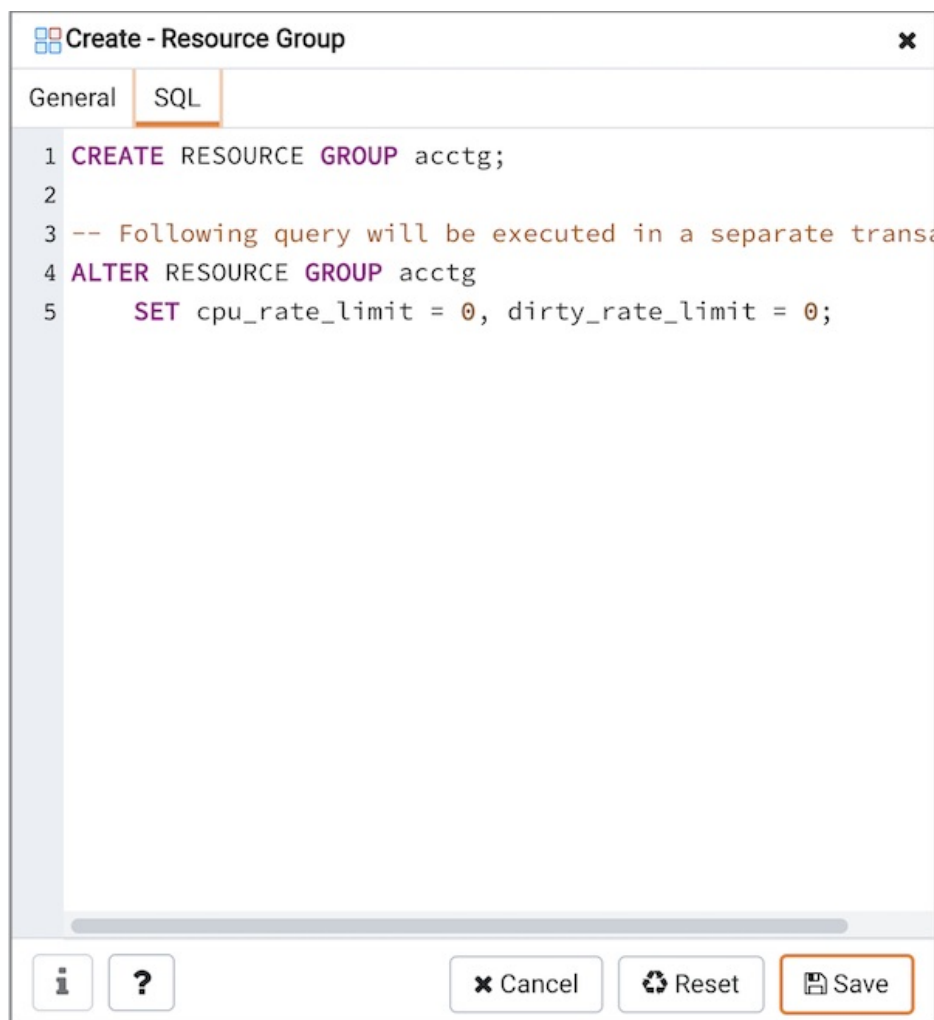
- Use the **Name** field to add a descriptive name for the resource group. This name will be displayed in the tree control.
- Use the **CPU rate limit (%)** field to set the value of the CPU rate limit resource type assigned to the resource group. The valid range for a CPU rate limit is from 0 to 1.67772e+07. The default value is 0.
- Use the **Dirty rate limit (KB)** field to set the value of the dirty rate limit resource type assigned to the resource group. The valid range for a dirty rate limit is from 0 to 1.67772e+07. The default value is 0.

Click the **SQL** tab to continue.

Your entries in the **Resource Group** dialog generate a SQL command. Use the **SQL** tab for review; revisit the **General** tab to make any changes to the SQL command.

#### Example

The following is an example of the sql command generated by selections made in the **Resource Group** dialog:



The example creates a resource group named `acctg` that sets `cpu_rate_limit` to `2`, and `dirty_rate_limit` to `6144`.

- Click the Info button ( **i** ) to access online SQL syntax reference material.
- Click the Help button ( **?** ) to access online documentation about Resource Groups.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

#### 34.5.10.4 Login/Group Role Dialog

Use the *Login/Group Role* dialog to define a role. A role may be an individual user (with or without login privileges) or a group of users. Note that roles defined at the cluster level are shared by all databases in the cluster.

The *Login/Group Role* dialog organizes the creation and management of roles through the following dialog tabs: *General*, *Definition*, *Privileges*, *Parameters*, and *Security*. The *SQL* tab displays the SQL code generated by dialog selections.

Create - Login/Group Role

General

Definition

Privileges

Membership

Parameters

Security

SQL

Name

alice

Comments

i

?

Cancel

Reset

Save

Use the fields on the *General* tab to identify the role.

- Use the *Name* field to provide the name of the role. The name will be displayed in the tree control.
- Provide a note about the role in the *Comments* field.

Click the *Definition* tab to continue.

Create - Login/Group Role

General

Definition

Privileges

Membership

Parameters

Security

SQL

Password

...

Account expires

YYYY-MM-DD HH:mm:ss Z

Connection limit

-1

i

?

Cancel

Reset

Save

Use the *Definition* tab to set a password and configure connection rules:

- Provide a password that will be associated with the role in the *Password* field.
- Provide an expiration date for the password in the *Account Expires* field (the role does not expire). The expiration date is not enforced when a user logs in with a non-password-based authentication method.
- If the role is a login role, specify how many concurrent connections the role can make in the *Connection Limit* field. The default value (-1) allows

unlimited connections.

Click the *Privileges* tab to continue.

The screenshot shows the 'Create - Login/Group Role' dialog box with the 'Privileges' tab selected. The dialog has tabs for General, Definition, Privileges, Membership, Parameters, Security, and SQL. The Privileges tab contains several toggle switches for role permissions:

| Privilege                                       | Value |
|---|-------|
| Can login?                                      | Yes   |
| Superuser?                                      | No    |
| Create roles?                                   | No    |
| Create databases?                               | No    |
| Update catalog?                                 | No    |
| Inherit rights from the parent roles?           | Yes   |
| Can initiate streaming replication and backups? | No    |

At the bottom of the dialog, there are buttons for 'Cancel', 'Reset', and 'Save' (highlighted with an orange border), along with information and help icons.

Use the *Privileges* tab to grant privileges to the role.

- Move the *Can login?* switch to the *Yes* position if the role has login privileges. The default value is *No*.
- Move the *Superuser?* switch to the *Yes* position if the role is a superuser within the database. The default value is *No*.
- Move the *Create roles?* switch to the *Yes* position to specify whether a role is permitted to create roles. A role with this privilege can alter and drop roles. The default value is *No*.
- Move the *Create databases* switch to the *Yes* position to control whether a role can create databases. The default value is *No*.
- The *Update catalog?* switch is disabled until the role is given superuser privileges. Move the *Update catalogs?* switch to the *No* position to control whether a role can update catalogs. The default value is *Yes* when the *Superuser?* switch is in the *Yes* position.
- Move the *Inherit rights from the parent roles?* switch to the *No* position if a role does not inherit privileges. The default value is *Yes*.
- Move the *Can initiate streaming replication and backups?* switch to the *Yes* position to control whether a role can initiate streaming replication or put the system in and out of backup mode. The default value is *No*.

Create - Login/Group Role

GeneralDefinitionPrivilegesMembershipParametersSecuritySQL

Roles

xpg\_monitor

Select the checkbox for roles to include WITH ADMIN OPTION.

i?

xCancelResetSave

- Specify member of the role in the *Member* offield and specify the members in the *Member*field. Confirm each selection by checking the checkbox to the right of the role name; delete a selection by clicking the *x* to the left of the role name. Membership conveys the privileges granted to the specified role to each of its members.

Click the *Parameters* tab to continue.

Create - Login/Group Role

GeneralDefinitionPrivilegesMembershipParametersSecuritySQL

| Name | Value | Database |
|------|-------|----------|
|------|-------|----------|

i?

xCancelResetSave

Use the fields on the *Parameters* tab to set session defaults for a selected configuration parameter when the role is connected to a specified database. This tab invokes the ALTER ROLE... SET configuration\_parameter syntax. Click the *Add* icon (+) to assign a value for a parameter.

- Use the drop-down listbox in the *Name* field to select a parameter.
- Use the *Value*field to specify a value for the parameter.
- Use the drop-down listbox in the *Database* field to select a database.



Click the *Add* icon (+) to specify each additional parameter; to discard a parameter, click the trash icon to the left of the row and confirm the deletion in the *Delete Row* popup.

Click the *Security* tab to continue.

Create - Login/Group Role

General

Definition

Privileges

Membership

Parameters

Security

SQL

Security labels

+

| Provider | Security label |
|----------|----------------|
|----------|----------------|

i

?

✕ Cancel

↺ Reset

💾 Save

Use the *Security* tab to define security labels applied to the role. Click the *Add* icon (+) to add each security label selection.

- Specify a security label provider in the *Provider* field. The named provider must be loaded and must consent to the proposed labeling operation.
- Specify a security label in the *Security Label* field. The meaning of a given label is at the discretion of the label provider. PostgreSQL places no restrictions on whether or how a label provider must interpret security labels; it merely provides a mechanism for storing them.

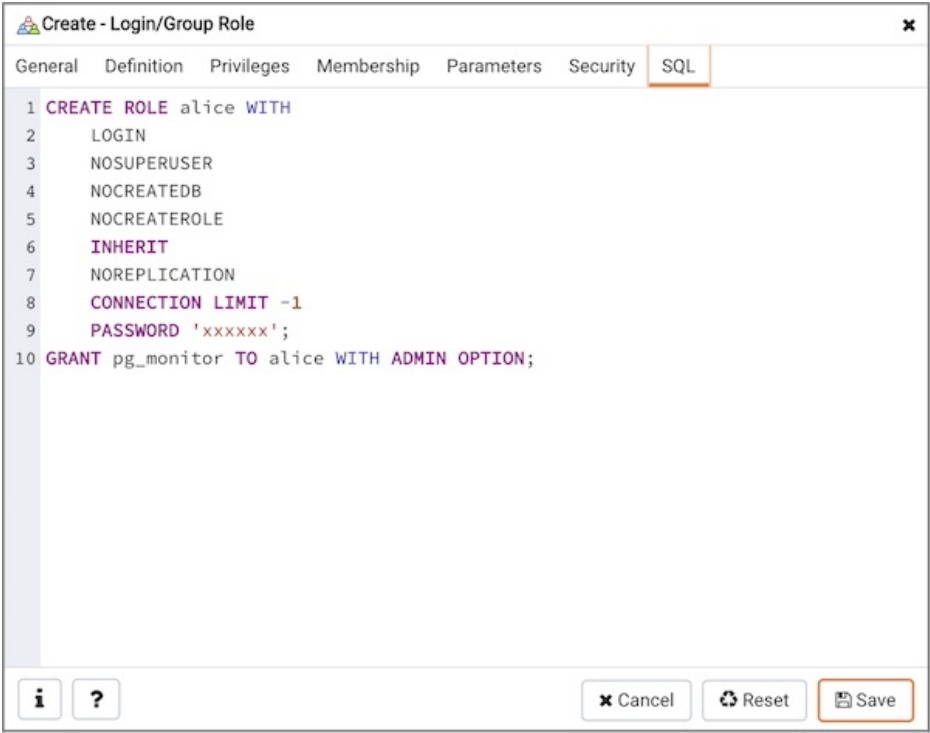
To discard a security label, click the trash icon to the left of the row and confirm the deletion in the *Delete Row* popup.

Click the *SQL* tab to continue.

Your entries in the *Login/Group Role* dialog generate a SQL command (see an example below). Use the *SQL* tab for review; revisit or switch tabs to make any changes to the SQL command.

Example

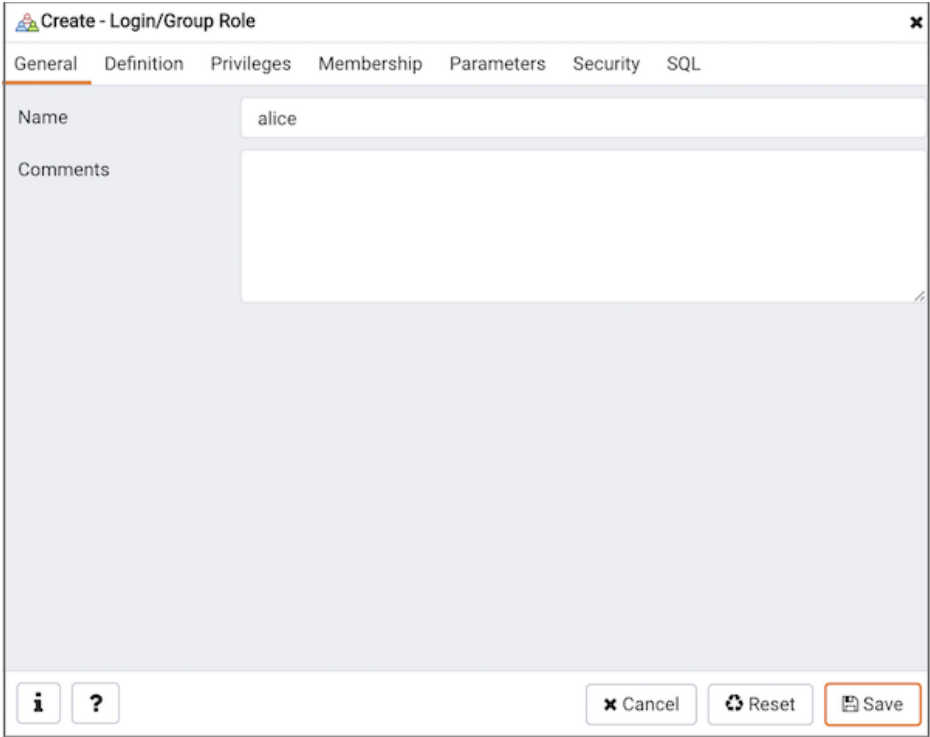
The following is an example of the sql command generated by user selections in the *Login/Group Role* dialog:



The example creates a login role named *alice* with *pem\_user* privileges; the role can make unlimited connections to the server at any given time.

- Click the Info button (i) to access online SQL help.
- Click the Help button (?) to access the documentation for the dialog.
- Click the Save button to save work.
- Click the Cancel button to exit without saving work.
- Click the Reset button to restore configuration parameters.

The **Login/Group Role** dialog organizes the creation and management of roles through the following dialog tabs: **General**, **Definition**, **Privileges**, **Parameters**, and **Security**. The **SQL** tab displays the SQL code generated by dialog selections.



Use the fields on the **General** tab to identify the role.

- Use the **Name** field to provide the name of the role. The name will be displayed in the tree control.
- Provide a note about the role in the **Comments** field.

Click the **Definition** tab to continue.

Create - Login/Group Role

GeneralDefinitionPrivilegesMembershipParametersSecuritySQL

Password

...

Account expires

YYYY-MM-DD HH:mm:ss Z

Connection limit

-1

i?

CancelResetSave

Use the **Definition** tab to set a password and configure connection rules:

- Provide a password that will be associated with the role in the **Password** field.
- Provide an expiration date for the password in the **Account Expires** field (the role does not expire). The expiration date is not enforced when a user logs in with a non-password-based authentication method.
- If the role is a login role, specify how many concurrent connections the role can make in the **Connection Limit** field. The default value (**-1**) allows unlimited connections.

Click the **Privileges** tab to continue.

Create - Login/Group Role

General

Definition

Privileges

Membership

Parameters

Security

SQL

Can login?

Yes

Superuser?

No

Create roles?

No

Create databases?

No

Update catalog?

No

Inherit rights from the parent roles?

Yes

Can initiate streaming replication and backups?

No

i

?

Cancel

Reset

Save

Use the **Privileges** tab to grant privileges to the role.

- Move the **Can login?** switch to the **Yes** position if the role has login privileges. The default value is **No**.
- Move the **Superuser** switch to the **Yes** position if the role is a superuser within the database. The default value is **No**.
- Move the **Create roles?** switch to the **Yes** position to specify whether a role is permitted to create roles. A role with this privilege can alter and drop roles. The default value is **No**.
- Move the **Create databases** switch to the **Yes** position to control whether a role can create databases. The default value is **No**.
- The **Update catalog?** switch is disabled until the role is given superuser privileges. Move the **Update catalogs?** switch to the **No** position to control whether a role can update catalogs. The default value is **Yes** when the **Superuser** switch is in the **Yes** position.
- Move the **Inherit rights from the parent roles?** switch to the **No** position if a role does not inherit privileges. The default value is **Yes**.
- Move the **Can initiate streaming replication and backups?** switch to the **Yes** position to control whether a role can initiate streaming replication or put the system in and out of backup mode. The default value is **No**.

Create - Login/Group Role

General

Definition

Privileges

Membership

Parameters

Security

SQL

Roles

x

pg\_monitor

x

Select the checkbox for roles to include WITH ADMIN OPTION.

i

?

Cancel

Reset

Save

- Specify members of the role in the **Role Membership** field. Click inside the **Roles** field to select role names from a drop down list. Confirm each selection by checking the checkbox to the right of the role name; delete a selection by clicking the **x** to the left of the role name. Membership conveys the privileges granted to the specified role to each of its members.

Click the **Parameters** tab to continue.

Create - Login/Group Role

GeneralDefinitionPrivilegesMembershipParametersSecuritySQL

| Name | Value | Database |
|------|-------|----------|
|------|-------|----------|

?

?

Cancel

Reset

Save

Use the fields on the **Parameters** tab to set session defaults for a selected configuration parameter when the role is connected to a specified database. This tab invokes the ALTER ROLE... SET configuration\_parameter syntax. Click the **Add** icon (+) to assign a value for a parameter.

- Use the drop-down listbox in the **Name** field to select a parameter.
- Use the **Value** field to specify a value for the parameter.
- Use the drop-down listbox in the **Database** field to select a database.

Click the **Add** icon (+) to specify each additional parameter; to discard a parameter, click the trash icon to the left of the row and confirm the deletion in the **Delete Row** popup.

Click the **Security** tab to continue.

Create - Login/Group Role

General Definition Privileges Membership Parameters **Security** SQL

Security labels +

| Provider | Security label |
|----------|----------------|
|----------|----------------|

Use the **Security** tab to define security labels applied to the role. Click the **Add** icon (+) to add each security label selection.

- Specify a security label provider in the **Provider** field. The named provider must be loaded and must consent to the proposed labeling operation.
- Specify a security label in the **Security Label** field. The meaning of a given label is at the discretion of the label provider. PostgreSQL places no restrictions on whether or how a label provider must interpret security labels; it merely provides a mechanism for storing them.

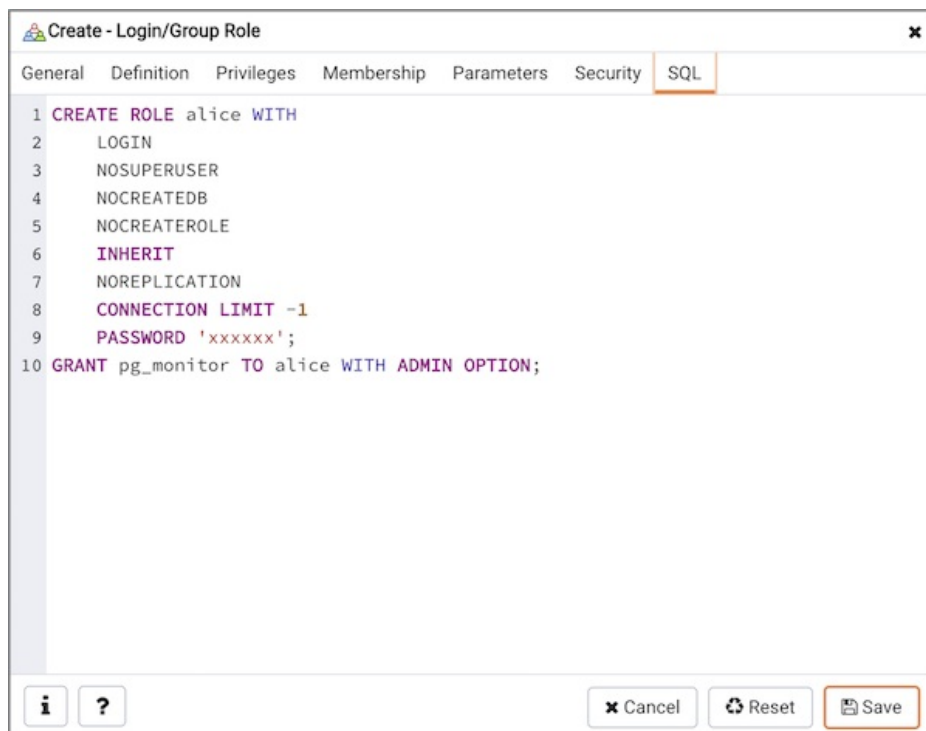
To discard a security label, click the trash icon to the left of the row and confirm the deletion in the **Delete Row** popup.

Click the **SQL** tab to continue.

Your entries in the **Login/Group Role** dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

### Example

The following is an example of the sql command generated by user selections in the **Login/Group Role** dialog:



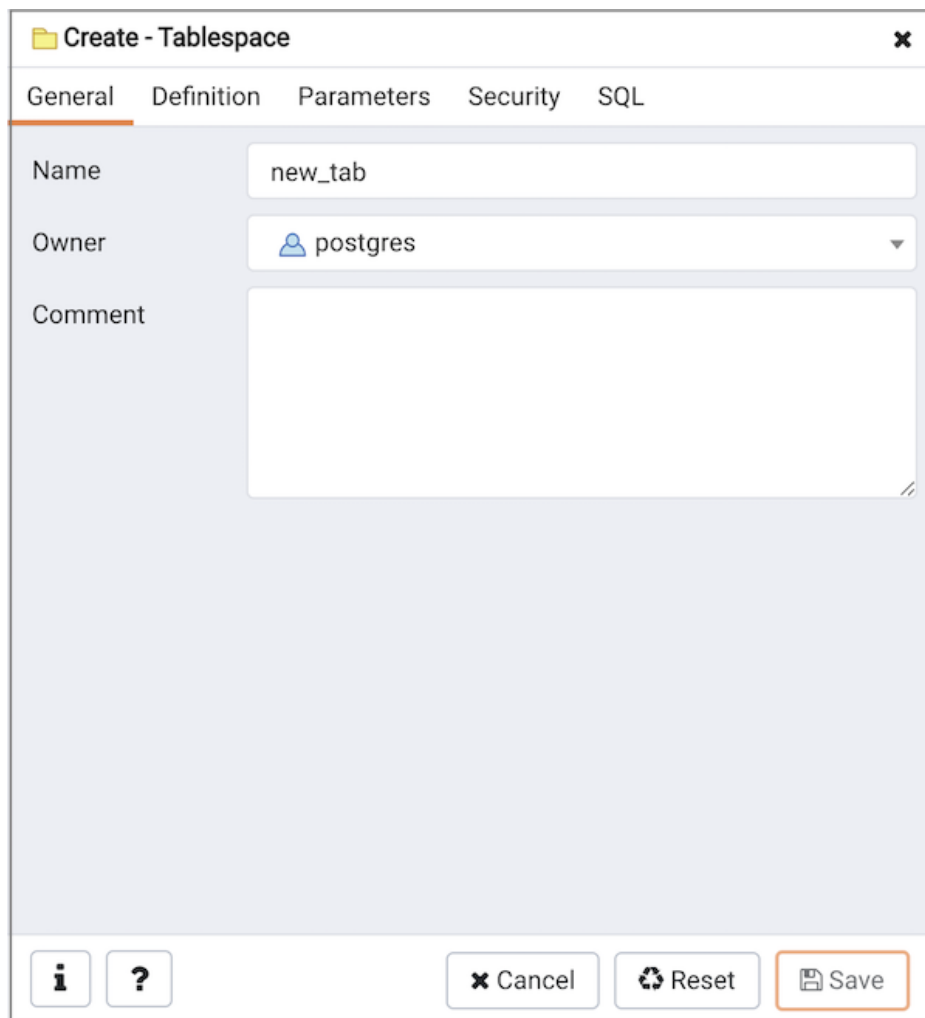
The example creates a login role named `alice` with `pg_monitor` privileges; the role can make unlimited connections to the server at any given time.

- Click the Info button ( `i` ) to access online SQL help.
- Click the Help button ( `?` ) to access the documentation for the dialog.
- Click the `Save` button to save work.
- Click the `Cancel` button to exit without saving work.
- Click the `Reset` button to restore configuration parameters.

### 34.5.10.5 Tablespace Dialog

Use The `Tablespace` dialog to define a tablespace. A tablespace allows superusers to define an alternative location on the file system where the data files containing database objects (such as tables and indexes) reside. Tablespaces are only supported on systems that support symbolic links. Note that a tablespace cannot be used independently of the cluster in which it is defined.

The `Tablespace` dialog organizes the definition of a tablespace through the following tabs: `General` , `Definition` , `Parameters` , and `Security` . The `SQL` tab displays the SQL code generated by dialog selections.

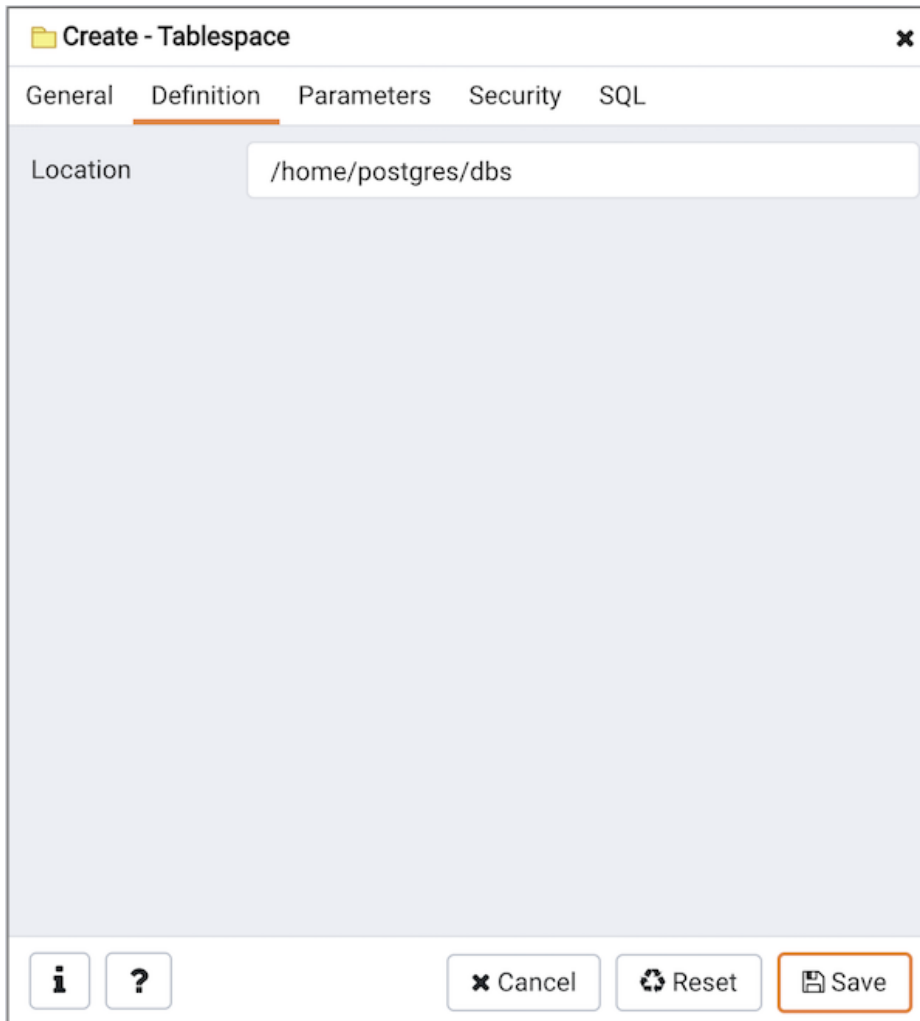


The image shows a 'Create - Tablespace' dialog box with a title bar and a close button. It has five tabs: 'General' (selected), 'Definition', 'Parameters', 'Security', and 'SQL'. The 'General' tab contains three fields: 'Name' with the value 'new\_tab', 'Owner' with a dropdown menu showing 'postgres' and a user icon, and 'Comment' with a large text area. At the bottom, there are four buttons: an information icon, a question mark icon, a 'Cancel' button, a 'Reset' button, and a 'Save' button (which is highlighted with an orange border).

- Use the **Name** field to identify the tablespace with a descriptive name. The name cannot begin with `pg_`; these names are reserved for system tablespaces.
- Select the owner of the tablespace from the drop-down listbox in the *Owner* field.
- Store notes about the tablespace in the **Comment** field.

Click the **Definition** tab to continue.





The image shows a 'Create - Tablespace' dialog box with a close button (X) in the top right corner. It has five tabs: 'General', 'Definition', 'Parameters', 'Security', and 'SQL'. The 'Definition' tab is selected and highlighted with an orange underline. Inside the 'Definition' tab, there is a 'Location' label and a text input field containing the path '/home/postgres/dbs'. At the bottom of the dialog, there are five buttons: an information icon (i), a help icon (?), a 'Cancel' button with an X icon, a 'Reset' button with a circular arrow icon, and a 'Save' button with a floppy disk icon. The 'Save' button is highlighted with an orange border.

- Use the **Location** field to specify an absolute path to a directory that will contain the tablespace.

Click the **Parameters** tab to continue.

**Create - Tablespace**

General Definition **Parameters** Security SQL

|   | Name               | Value |
|---|--------------------|-------|
| 🗑 | random_page_cost ▼ | 1     |

ⓘ ? 
 ✕ Cancel
🔄 Reset
💾 Save

Use the **Parameters** tab to set parameters for the tablespace. Click the **Add** icon (+) to add a row to the table below.

- Use the drop-down listbox next to **Name** to select a parameter.
- Use the **Value** field to set a value for the parameter.

Click the **Add** icon (+) to specify each additional parameter; to discard a parameter, click the trash icon to the left of the row and confirm deletion in the **Delete Row** dialog.

Click the **Security** tab to continue.

**Create - Tablespace**

General Definition Parameters **Security** SQL

**Privileges** +

| Grantee | Grantor | Privileges |
|---------|---------|------------|
|---------|---------|------------|

**Security labels** +

| Provider | Security label |
|----------|----------------|
|----------|----------------|

Use the **Security** tab to assign privileges and define security labels for the tablespace.

Use the **Privileges** panel to assign security privileges. Click the **Add** icon (+) to assign a set of privileges:

- Select the name of the role from the drop-down listbox in the **Grantee** field.
- The current user, who is the default grantor for granting the privilege, is displayed in the **Grantor** field.
- Click inside the **Privileges** field. Check the boxes to the left of one or more privileges to grant the selected privileges to the specified user.

Click the **Add** icon to assign additional sets of privileges; to discard a privilege, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Use the **Security Labels** panel to define security labels applied to the tablespace. Click the **Add** icon (+) to add each security label selection:

- Specify a security label provider in the **Provider** field. The named provider must be loaded and must consent to the proposed labeling operation.
- Specify a security label in the **Security Label** field. The meaning of a given label is at the discretion of the label provider. PostgreSQL places no restrictions on whether or how a label provider must interpret security labels; it merely provides a mechanism for storing them.

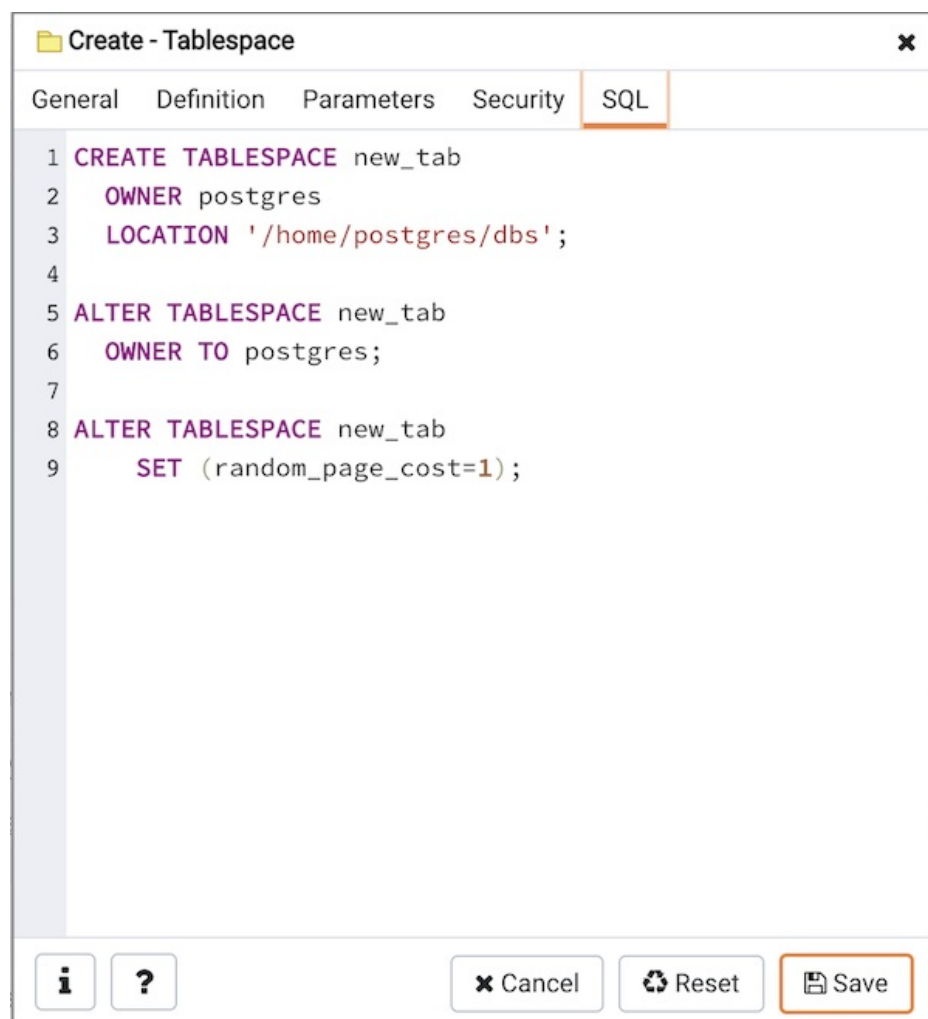
To discard a security label, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Click the **SQL** tab to continue.

Your entries in the **Tablespace** dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

## Example

The following is an example of the sql command generated by user selections in the **Tablespace** dialog:



The example shown demonstrates creating a tablespace named **space\_01**. It has a *random\_page\_cost* value equal to **1**.

- Click the **Info** button (i) to access online help.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

### 34.5.10.6 Role Reassign/Drop Own Dialog

Use the *Reassign/Drop Own* dialog to change the ownership of database objects owned by a database role. This dialog instructs the system to change the ownership of database objects owned by any of the *old\_roles* to *new\_role*.

The *Reassign/Drop Own* dialog organizes the Reassign & Drop role through General tab.

Reassign/Drop Owned - 'agent1'

General
SQL

Operation

✓ Reassign

Drop

Change the ownership or drop the database objects owned by a database role

Reassign objects to

CURRENT\_USER

New owner of the affected objects

Cascade?

No

Note: CASCADE will automatically drop objects that depend on the affected objects, and in turn all objects that depend on those objects

From database

pem

Target database on which the operation will be carried out

?

Cancel
OK

- Use the *Operation* field to provide Reassign option.
- Provide a new role in the *Reassign Objects to* field; The ownership of all the objects within the selected database, and of all shared objects (databases, tablespaces), owned by the *old\_role* will be reassigned to *new\_role*.
- Provide a database on which the reassignment is to be carried out.

The above example demonstrates reassigning *old\_role* to *new\_role*.

Click the *SQL* tab to continue.

Reassign/Drop Owned - 'agent1'

General SQL

1 REASSIGN OWNED BY agent1 TO CURRENT\_USER

? × Cancel OK

Removing database objects owned by a database role.

Reassign/Drop Owned - 'agent1'

General

SQL

Operation

Reassign

✓ Drop

Change the ownership or drop the database objects owned by a database role

Reassign objects to


New owner of the affected objects

Cascade?

No

Note: CASCADE will automatically drop objects that depend on the affected objects, and in turn all objects that depend on those objects

From database

 pem

Target database on which the operation will be carried out

?

✕ Cancel

OK

- Use the *Operation* field to provide Drop option.
- Use the *Cascade?* field to provide Yes, No is default.
- Provide a database on which the drop of objects is to be carried out.

Click the *SQL* tab to continue.



The above examples demonstrates drop owned by *role*.

- Click the *Help* button (?) to access online help.
- Click the *OK* button to save work.
- Click the *Cancel* button to exit without saving work.

## 34.5.11 Managing Database Objects

PEM provides simple but powerful dialogs that you can use to design and create database objects. Each dialog contains a series of tabs that you use to describe the object that will be created by the dialog; the SQL tab displays the SQL command that the server will execute when creating the object.

To access a dialog that allows you to create a database object, right-click on the object type in the Browser tree control, and select the **Create** option for that object. For example, to create a new cast, right-click on the **Casts** node, and select **Create Cast...**

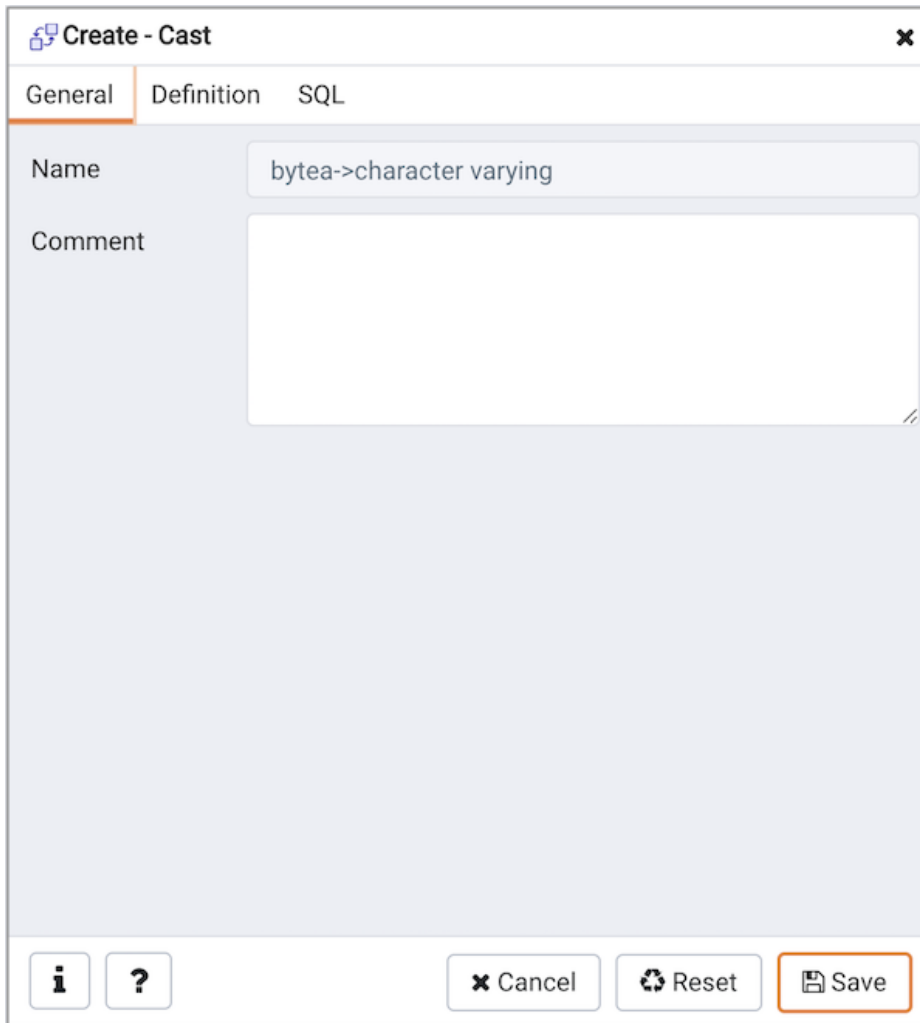
Contents:

### 34.5.11.1 Cast Dialog

Use the **Cast** dialog to define a cast. A cast specifies how to convert a value from one data type to another.

The **Cast** dialog organizes the development of a cast through the following dialog tabs: **General** and **Definition**. The **SQL** tab displays the SQL code generated by dialog selections.





The image shows a 'Create - Cast' dialog box with three tabs: 'General', 'Definition', and 'SQL'. The 'General' tab is selected. It contains a 'Name' field with the text 'bytea->character varying' and a 'Comment' text area. At the bottom, there are buttons for 'Cancel', 'Reset', and 'Save', along with information and help icons.

Create - Cast

General Definition SQL

Name bytea->character varying

Comment

Cancel Reset Save

Use the fields in the **General** tab to identify the cast:

- The **Name** field is disabled. The name that will be displayed in the **Browser** tree control is the **Source** type concatenated with the **Target** type, and is generated automatically when you make selections on the **Cast** dialog **Definition** tab.
- Store notes about the cast in the **Comment** field.

Click the **Definition** tab to continue.

**Create - Cast**

General **Definition** SQL

Source type: bytea

Target type: character varying

Function: Select from the list

Context: ☒ EXPLICIT

Buttons: [i] [?] [x] Cancel [Reset] [Save]

Use the fields in the **Definition** tab to define parameters:

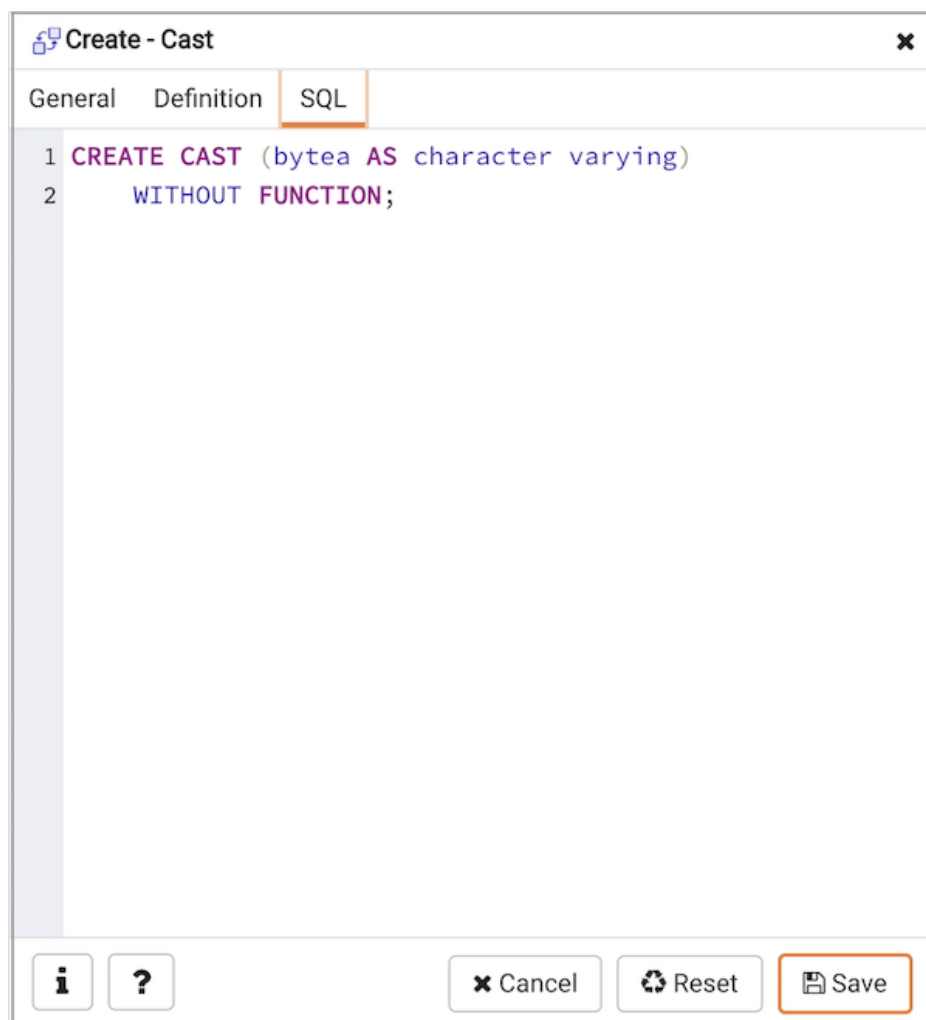
- Use the drop-down listbox next to **Source type** to select the name of the source data type of the cast.
- Use the drop-down listbox next to **Target type** to select the name of the target data type of the cast.
- Use the drop-down listbox next to **Function** to select the function used to perform the cast. The function's result data type must match the target type of the cast.
- Move the **Context** switch to the **Implicit** position if the cast is implicit. By default, a cast can be invoked only by an explicit cast request. If the cast is marked **Implicit** then it can be invoked implicitly in any context, whether by assignment or internally in an expression.

Click the **SQL** tab to continue.

Your entries in the **Cast** dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

#### Example

The following is an example of the sql command generated by user selections in the **Cast** dialog:



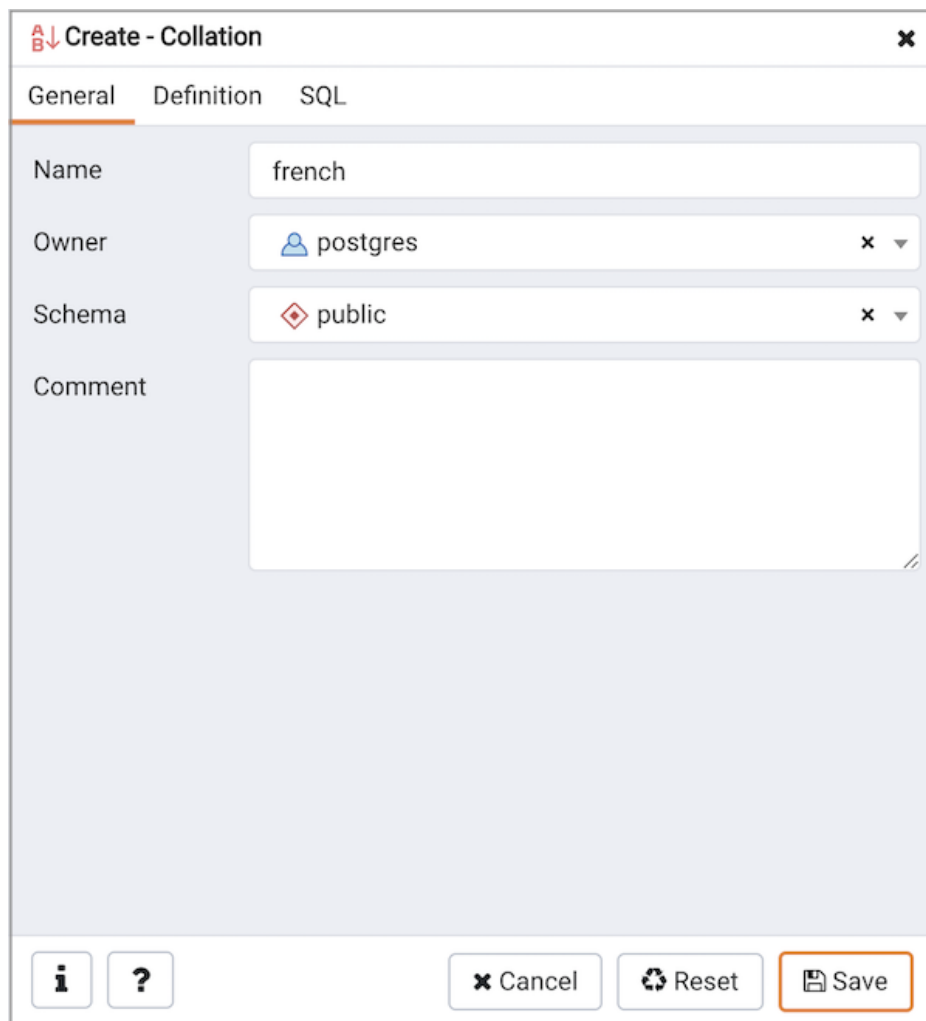
The cast uses a function named `int4(bigint)` to convert a bigint data type to an integer.

- Click the **Info** button (i) to access online help. View context-sensitive help in the **Tabbed browser**, where a new tab displays the PostgreSQL core documentation.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

### 34.5.11.2 Collation Dialog

Use the **Collation** dialog to define a collation. A collation is an SQL schema object that maps a SQL name to operating system locales. To create a collation, you must have a CREATE privilege on the destination schema.

The **Collation** dialog organizes the development of a collation through the following dialog tabs: **General** and **Definition**. The **SQL** tab displays the SQL code generated by dialog selections.



**Create - Collation**

General Definition SQL

Name: french

Owner: postgres

Schema: public

Comment:

Buttons: Cancel, Reset, Save

Use the fields in the **General** tab to identify the collation:

- Use the **Name** field to provide a name for the collation. The collation name must be unique within a schema. The name will be displayed in the **Browser** tree control.
- Select the name of the owner from the drop-down listbox in the **Owner** field.
- Select the name of the schema in which the collation will reside from the drop-down listbox in the **Schema** field.
- Store notes about the collation in the **Comment** field.

Click the **Definition** tab to continue.

**Create - Collation**

General **Definition** SQL

Copy collation

Locale

LC\_COLLATE

LC\_TYPE

**i** **?** **Cancel** **Reset** **Save**

Use the fields in the **Definition** tab to specify the operating system locale settings:

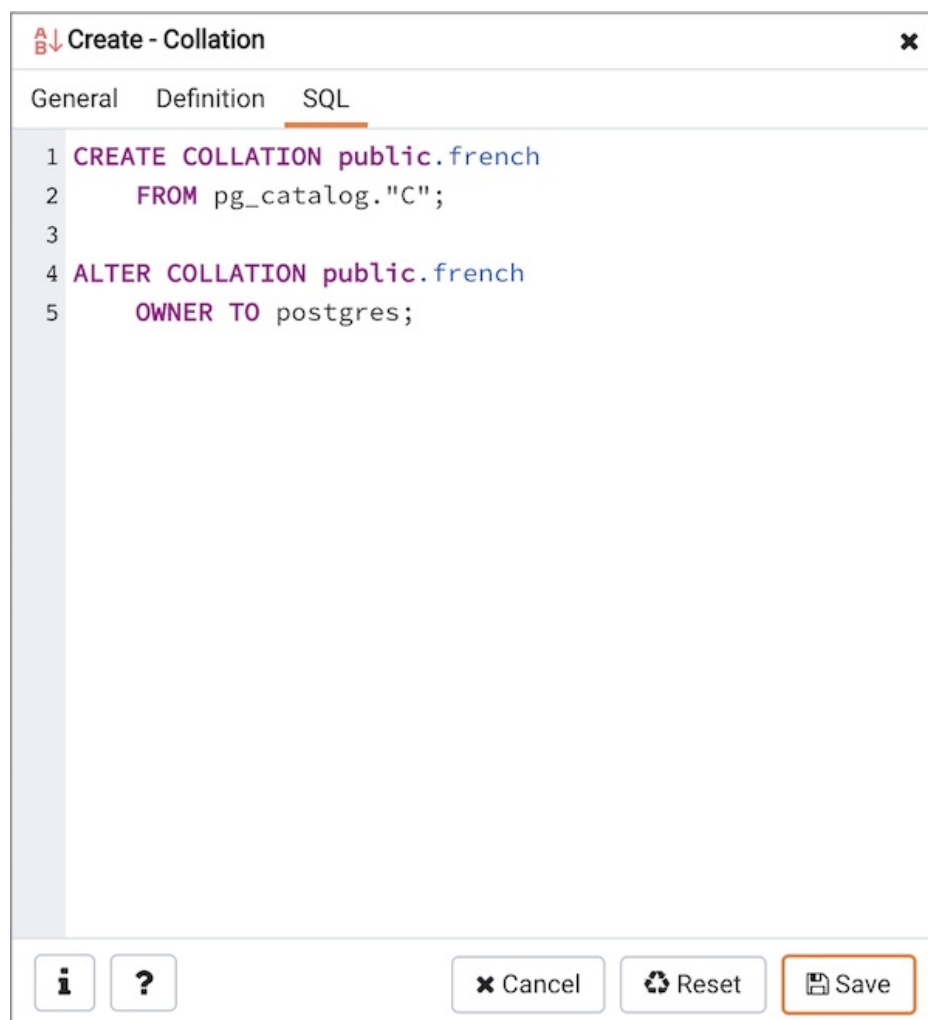
- Use the drop-down listbox next to **Copy collation** to select the name of an existing collation to copy. The new collation will have the same properties as the existing one, but will be an independent object. If you choose to copy an existing collation, you cannot modify the collation properties displayed on this tab.
- Use the **Locale** field to specify a locale; a locale specifies language and language formatting characteristics. If you specify this, you cannot specify either of the following parameters. To view a list of locales supported by your Linux system use the command `locale -a`.
- Use the **LC\_COLLATE** field to specify a locale with specified string sort order. The locale must be applicable to the current database encoding. (See CREATE DATABASE for details.)
- Use the **LC\_CTYPE** field to specify a locale with specified character classification. The locale must be applicable to the current database encoding. (See CREATE DATABASE for details.)

Click the **SQL** tab to continue.

Your entries in the **Collation** dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

## Example

The following is an example of the sql command generated by user selections in the **Collation** dialog:



The example shown demonstrates creating a collation named `french` that uses the rules specified for the locale, `fr-BI-x-icu`. The collation is owned by `postgres`.

- Click the **Info** button (i) to access online help. For more information about setting a locale, see Chapter 22.1 Locale Support of the PostgreSQL core documentation:

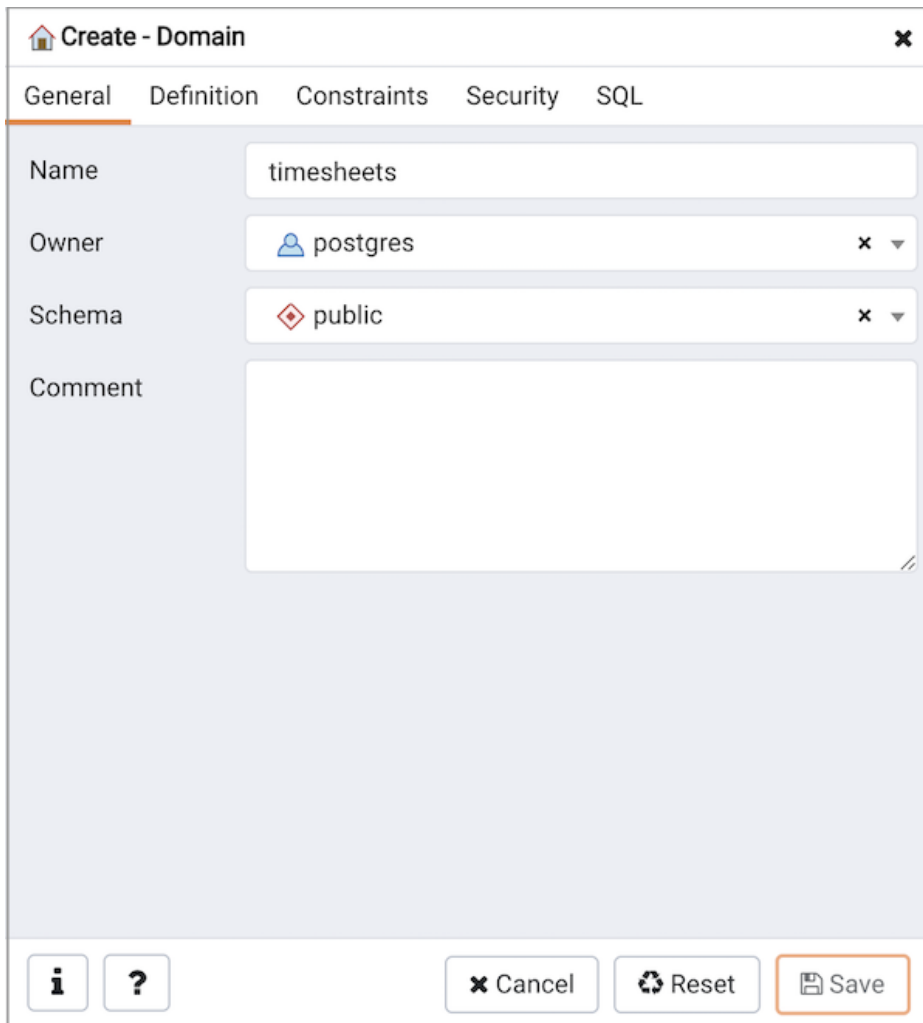
<http://www.postgresql.org/docs/current/static/locale.html>

- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

### 34.5.11.3 Domain Dialog

Use the **Domain** dialog to define a domain. A domain is a data type definition that may constrain permissible values. Domains are useful when you are creating multiple tables that contain comparable columns; you can create a domain that defines constraints that are common to the columns and re-use the domain definition when creating the columns, rather than individually defining each set of constraints.

The **Domain** dialog organizes the development of a domain through the following tabs: **General**, **Definition**, **Constraints**, and **Security**. The **SQL** tab displays the SQL code generated by dialog selections.



**Create - Domain**

General Definition Constraints Security SQL

Name: timesheets

Owner: postgres

Schema: public

Comment:

Buttons: [i] [?] [Cancel] [Reset] [Save]

Use the fields on the **General** tab to identify a domain:

- Use the **Name** field to add a descriptive name for the domain. The name will be displayed in the **Browser** tree control.
- Use the drop-down listbox next to **Owner** to select a role that will own the domain.
- Select the name of the schema in which the domain will reside from the drop-down listbox in the **Schema** field.
- Store notes about the domain in the **Comment** field.

Click the **Definition** tab to continue.

**Create - Domain**

General **Definition** Constraints Security SQL

Base type: character varying

Length:

Precision:

Default: Enter an expression or a value.

Not NULL?: ☐ No

Collation: Select from the list

i ? Cancel Reset Save

Use the fields in the **Definition** tab to describe the domain:

- Use the drop-down listbox next to **Base type** to specify a data type.
- Use the context-sensitive **Length** field to specify a numeric length for a numeric type.
- Use the context-sensitive **Precision** field to specify the total count of significant digits for a numeric type.
- Specify a default value for the domain data type in the **Default** field. The data type of the default expression must match the data type of the domain. If no default value is specified, then the default value is the null value.
- Move the **Not Null** switch to specify the values of this domain are prevented from being null.
- Use the drop-down listbox next to **Collation** to apply a collation cast. If no collation is specified, the underlying data type's default collation is used. The underlying type must be collatable if COLLATE is specified.

Click the **Constraints** tab to continue.



Create - Domain

General

Definition

Constraints

Security

SQL

Constraints

+

|             | Name | Check                                     | Validate?   |
|-------------|------|---|-------------|
| <div></div> | chk  | value in ('Mon','Tue','Wed','Thur','Fri') | <div></div> |

i

?

Cancel

Reset

Save

Use the fields in the **Constraints** tab to specify rules for the domain. Click the **Add** icon (+) to set constraints:

- Use the **Name** field to specify a name for the constraint.
- Use the **Check** field to provide an expression for the constraint.
- Use the **Validate** checkbox to determine whether the constraint will be validated. The default checkbox is checked and sets a validation requirement.

A CHECK clause specifies an integrity test which values of the domain must satisfy. Each constraint must be an expression that produces a Boolean result. Use the key word VALUE to refer to the value being tested. Expressions evaluating to TRUE or UNKNOWN succeed. If the expression produces a FALSE result, an error is reported and the value is not allowed to be converted to the domain type. A CHECK expression cannot contain subqueries nor refer to variables other than VALUE. If a domain has multiple CHECK constraints, they will be tested in alphabetical order by name.

Click the **Add** icon (+) to set additional constraints; to discard a constraint, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Click the **Security** tab to continue.

Create - Domain

General

Definition

Constraints

Security

SQL

Security labels

+

| Provider | Security label |
|----------|----------------|
|----------|----------------|

i

?

✕ Cancel

↺ Reset

💾 Save

Use the **Security Labels** panel to assign security labels. Click the **Add** icon (+) to add a label:

- Specify a security label provider in the **Provider** field. The named provider must be loaded and must consent to the proposed labeling operation.
- Specify a security label in the **Security Label** field. The meaning of a given label is at the discretion of the label provider. PostgreSQL places no restrictions on whether or how a label provider must interpret security labels; it merely provides a mechanism for storing them.

Click the **Add** icon (+) to specify each additional label; to discard a label, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Click the **SQL** tab to continue.

Your entries in the **Domain** dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

Example

The following is an example of the sql command generated by selections made in the **Domain** dialog:



The example shown demonstrates creating a domain named `minimum-wage` that confirms that the value entered is greater than or equal to `7.25`.

- Click the **Info** button (i) to access online help. View context-sensitive help in the **Tabbed browser**, where a new tab displays the PostgreSQL core documentation.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

#### 34.5.11.4 Domain Constraints Dialog

Use the **Domain Constraints** dialog to create or modify a domain constraint. A domain constraint confirms that the values provided for a domain meet a defined criteria. The **Domain Constraints** dialog implements options of the ALTER DOMAIN command.

The **Domain Constraints** dialog organizes the development of a domain constraint through the following dialog tabs: **General** and **Definition**. The **SQL** tab displays the SQL code generated by dialog selections.



The image shows a 'Create - Domain Constraints' dialog box with three tabs: 'General', 'Definition', and 'SQL'. The 'General' tab is selected. It contains a 'Name' field with the value 'chk' and a 'Comment' field which is empty. At the bottom, there are buttons for 'Cancel', 'Reset', and 'Save' (highlighted with an orange border), along with information and help icons.

Create - Domain Constraints

General Definition SQL

Name: chk

Comment:

Buttons: [i] [?] [x] Cancel [Reset] [Save]

Use the fields in the **General** tab to identify the domain constraint:

- Use the **Name** field to add a descriptive name for the constraint. The name will be displayed in the **Browser** tree control.
- Store notes about the constraint in the **Comment** field.

Click the **Definition** tab to continue.

**Create - Domain Constraints**

General **Definition** SQL

Check: value in ('Mon','Tue','Wed','Thur','Fri')

Validate? ☒ Yes

Buttons: [i] [?] [Cancel] [Reset] [Save]

Use the fields in the **Definition** tab to define the domain constraint:

- Use the **Check** field to provide a CHECK expression. A CHECK expression specifies a constraint that the domain must satisfy. A constraint must produce a Boolean result; include the key word VALUE to refer to the value being tested. Only those expressions that evaluate to TRUE or UNKNOWN will succeed. A CHECK expression cannot contain subqueries or refer to variables other than VALUE. If a domain has multiple CHECK constraints, they will be tested in alphabetical order.
- Move the **Validate?** switch to the **No** position to mark the constraint NOT VALID. If the constraint is marked NOT VALID, the constraint will not be applied to existing column data. The default value is **Yes**.

Click the **SQL** tab to continue.

Your entries in the **Domain Constraints** dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

#### Example

The following is an example of the sql command generated by user selections in the **Domain Constraints** dialog:



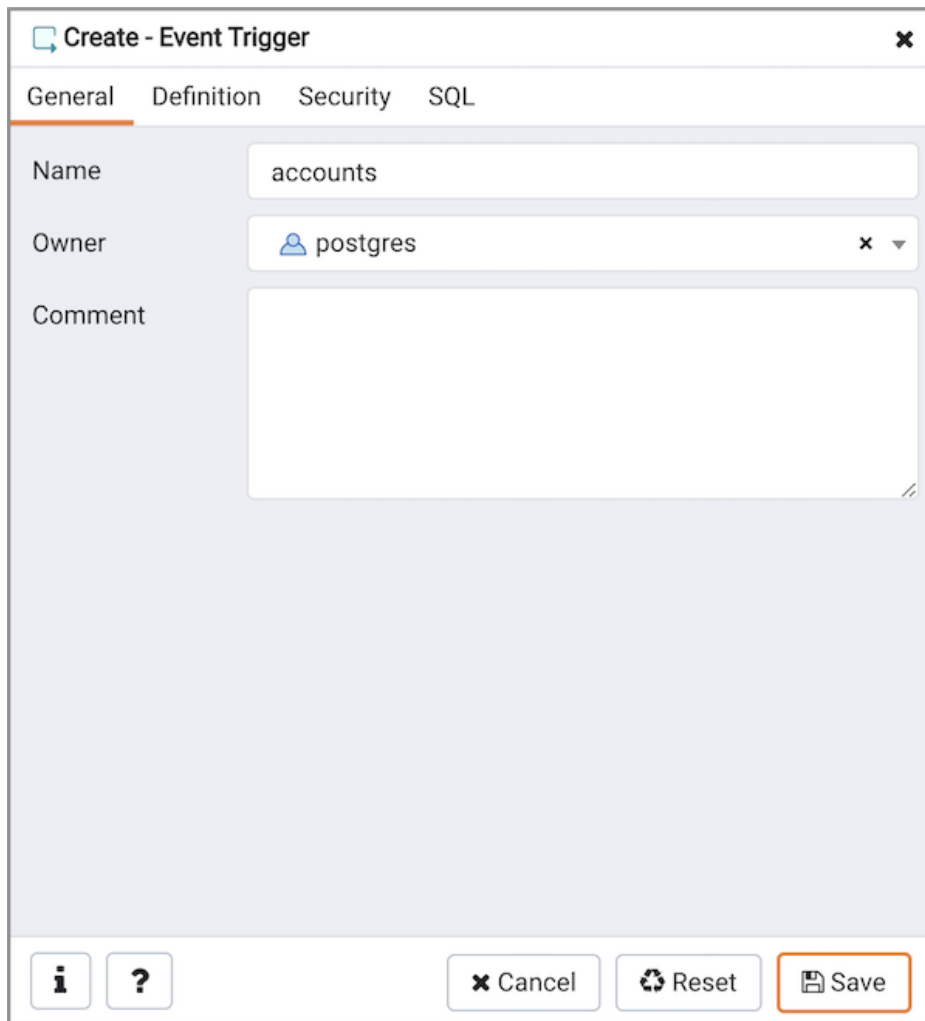
The example shown demonstrates creating a domain constraint on the domain `timesheets` named `weekday`. It constrains a value to equal `Friday`.

- Click the **Info** button (i) to access online help. View context-sensitive help in the **Tabbed browser**, where a new tab displays the PostgreSQL core documentation.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

### 34.5.11.5 Event Trigger Dialog

Use the **Domain Trigger** dialog to define an event trigger. Unlike regular triggers, which are attached to a single table and capture only DML events, event triggers are global to a particular database and are capable of capturing DDL events. Like regular triggers, event triggers can be written in any procedural language that includes event trigger support, or in C, but not in SQL.

The **Domain Trigger** dialog organizes the development of a event trigger through the following dialog tabs: **General**, **Definition**, and **Security Labels**. The **SQL** tab displays the SQL code generated by dialog selections.



The image shows a 'Create - Event Trigger' dialog box with four tabs: General, Definition, Security, and SQL. The 'General' tab is selected. It contains three fields: 'Name' with the value 'accounts', 'Owner' with a dropdown menu showing 'postgres', and a large 'Comment' text area. At the bottom, there are buttons for 'Cancel', 'Reset', and 'Save' (which is highlighted with an orange border). There are also information and help icons on the left.

Create - Event Trigger

General Definition Security SQL

Name accounts

Owner postgres

Comment

Cancel Reset Save

Use the fields in the **General** tab to identify the event trigger:

- Use the **Name** field to add a descriptive name for the event trigger. The name will be displayed in the **Browser** tree control.
- Use the drop-down listbox next to **Owner** to specify the owner of the event trigger.
- Store notes about the event trigger in the **Comment** field.

Click the **Definition** tab to continue.

**Create - Event Trigger**

General **Definition** Security SQL

Trigger enabled? Enable

Trigger function test\_func

Event DDL COMMAND START

When TAG in 1

**?** **?** **Cancel** **Reset** **Save**

Use the fields in the **Definition** tab to define the event trigger:

- Select a value from the drop down of **Trigger Enabled** field to specify a status Select a value from the drop down of **Trigger Enabled** field to specify a status
- Use the drop-down listbox next to **Trigger function** to specify an existing function. A trigger function takes an empty argument list, and returns a value of type event\_trigger.
- Select a value from the drop down of **Events** field to specify when the event trigger will fire: **DDL COMMAND START** , **DDL COMMAND END** , or **SQL DROP** .
- Use the **When TAG in** field to enter filter values for TAG for which the trigger will be executed. The values must be in single quotes separated by comma.

Click the **Security Labels** tab to continue.



Create - Event Trigger

General

Definition

Security

SQL

Security labels

| Provider | Security label |
|----------|----------------|
|----------|----------------|

i

?

Cancel

Reset

Save

Use the **Security** tab to define security labels applied to the trigger. Click the **Add** icon (+) to add each security label.

- Specify a security label provider in the **Provider** field. The named provider must be loaded and must consent to the proposed labeling operation.
- Specify a security label in the **Security Label** field. The meaning of a given label is at the discretion of the label provider. PostgreSQL places no restrictions on whether or how a label provider must interpret security labels; it merely provides a mechanism for storing them.

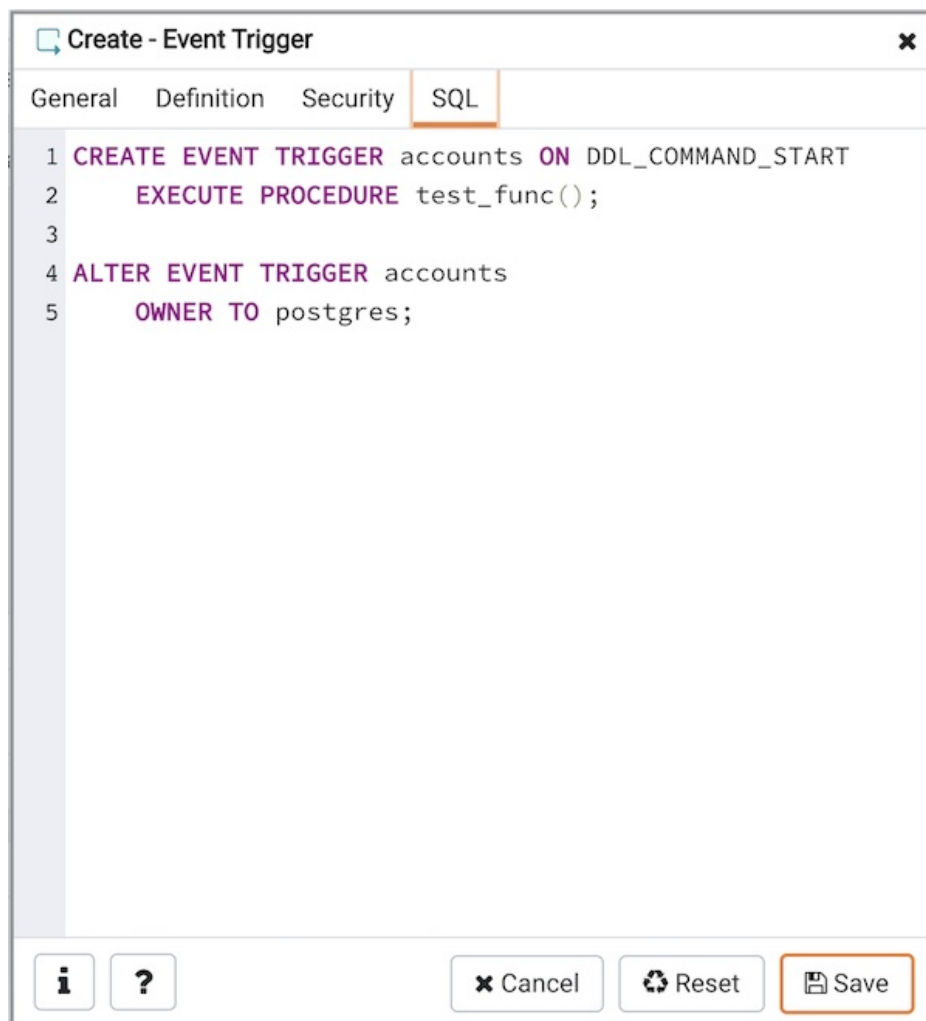
Click the **Add** icon (+) to assign additional security labels; to discard a security label, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Click the **SQL** tab to continue.

Your entries in the **Domain Trigger** dialog generate a generate a SQL command. Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

Example

The following is an example of the sql command generated by user selections in the **Domain Trigger** dialog:



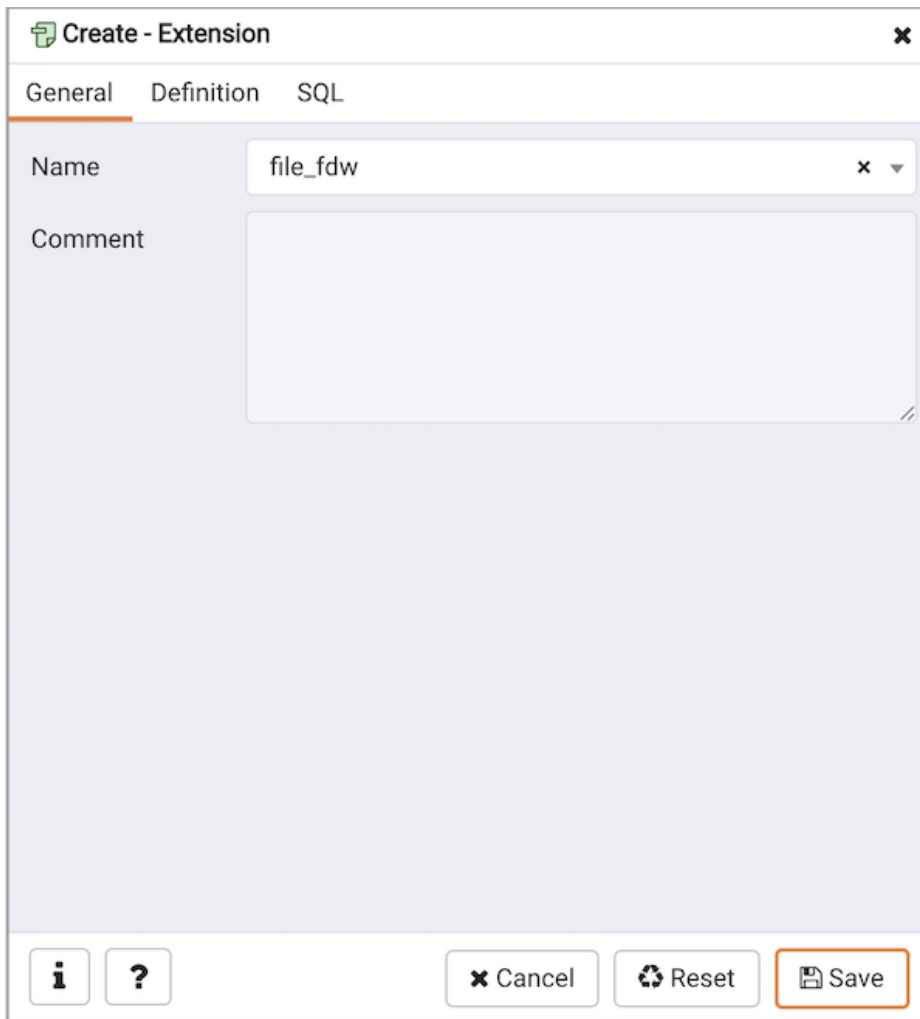
The command creates an event trigger named `accounts` that invokes the procedure named `acct_due`.

- Click the **Info** button (i) to access online help. View context-sensitive help in the **Tabbed browser**, where a new tab displays the PostgreSQL core documentation.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

### 34.5.11.6 Extension Dialog

Use the **Extension** dialog to install a new extension into the current database. An extension is a collection of SQL objects that add targeted functionality to your Postgres installation. The **Extension** dialog adds the functionality of an extension to the current database only; you must register the extension in each database that use the extension. Before you load an extension into a database, you should confirm that any pre-requisite files are installed.

The **Extension** dialog allows you to implement options of the CREATE EXTENSION command through the following dialog tabs: **General** and **Definition**. The **SQL** tab displays the SQL code generated by dialog selections.



The image shows a 'Create - Extension' dialog box with three tabs: 'General', 'Definition', and 'SQL'. The 'General' tab is active. It contains a 'Name' field with a dropdown menu showing 'file\_fdw' and a 'Comment' text area. At the bottom, there are buttons for 'Cancel', 'Reset', and 'Save', along with information and help icons.

Create - Extension

General Definition SQL

Name file\_fdw

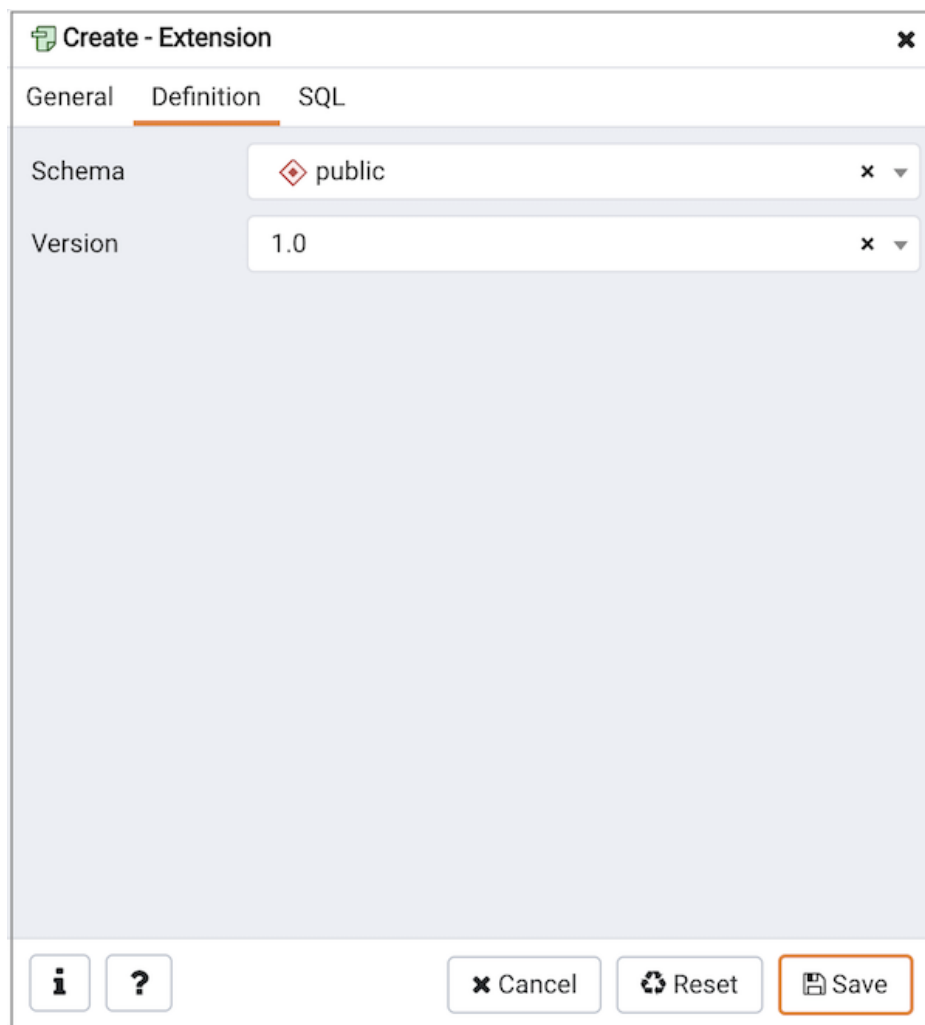
Comment

Cancel Reset Save

Use the fields in the **General** tab to identify an extension:

- Use the drop-down listbox in the **Name** field to select the extension. Each extension must have a unique name.
- Store notes about the extension in the **Comment** field.

Click the **Definition** tab to continue.



The image shows a 'Create - Extension' dialog box with three tabs: 'General', 'Definition', and 'SQL'. The 'Definition' tab is selected. It contains two fields: 'Schema' with a dropdown menu showing 'public' and 'Version' with a dropdown menu showing '1.0'. At the bottom, there are buttons for 'Cancel', 'Reset', and 'Save', along with information and help icons.

Use the **Definition** tab to select the **Schema** and **Version**:

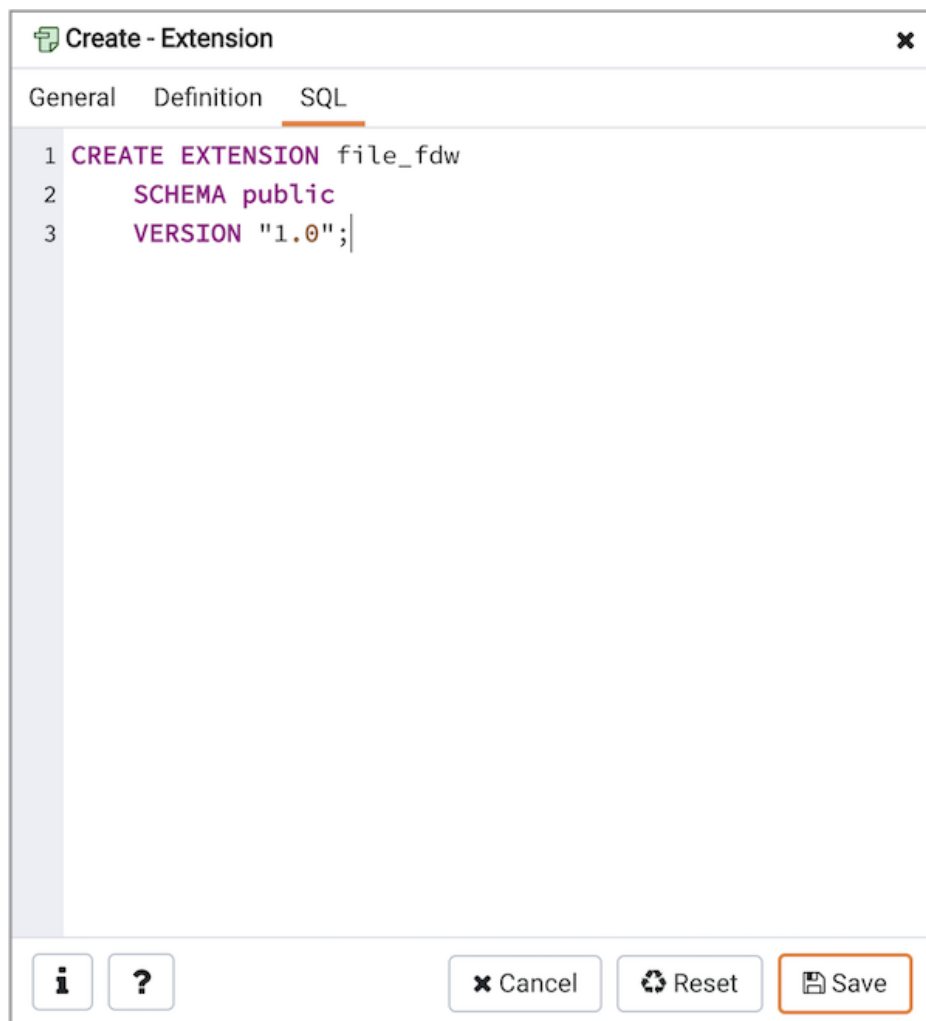
- Use the drop-down listbox next to **Schema** to select the name of the schema in which to install the extension's objects.
- Use the drop-down listbox next to **Version** to select the version of the extension to install.

Click the **SQL** tab to continue.

Your entries in the **Extension** dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

#### Example

The following is an example of the sql command generated by user selections in the **Extension** dialog:



The command creates the `chkpass` extension in the `public` schema. It is version `1.0` of `chkpass`.

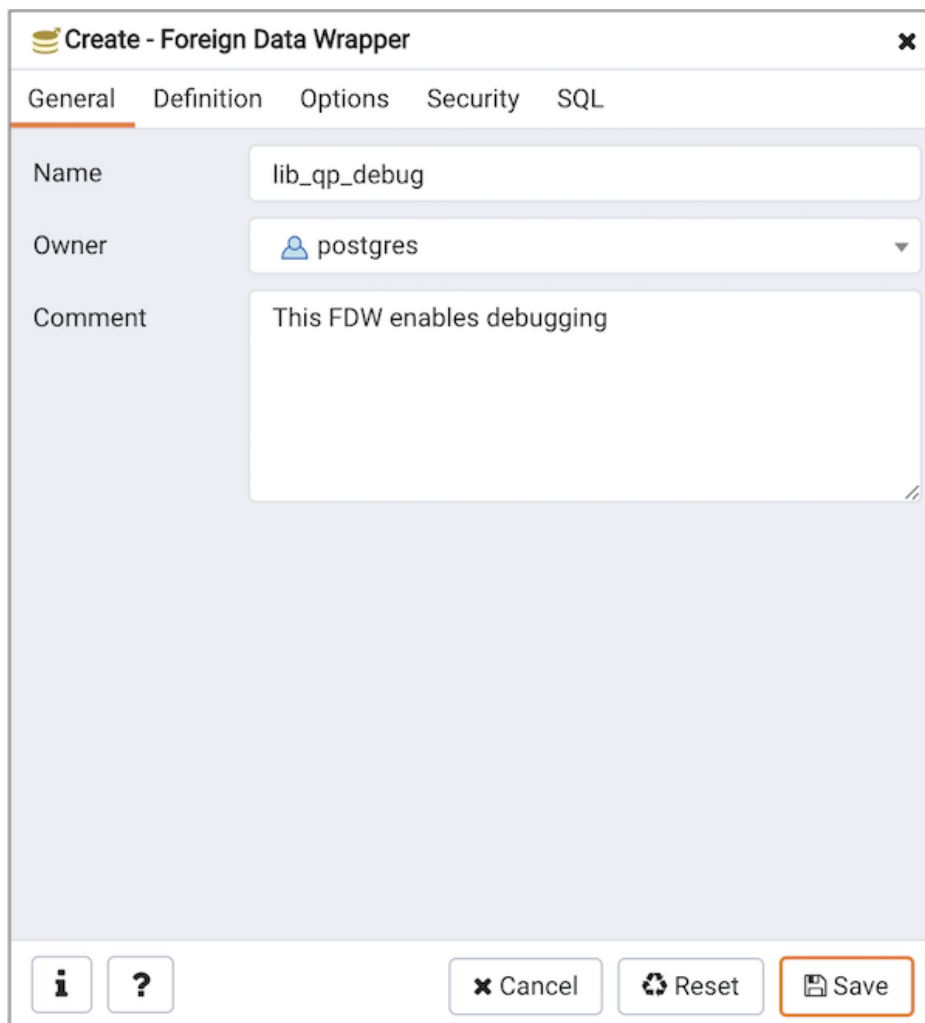
- Click the **Info** button (i) to access online help. View context-sensitive help in the **Tabbed browser**, where a new tab displays the PostgreSQL core documentation.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

### 34.5.11.7 Foreign Data Wrapper Dialog

Use the **Foreign Data Wrapper** dialog to create or modify a foreign data wrapper. A foreign data wrapper is an adapter between a Postgres database and data stored on another data source.

You must be a superuser to create a foreign data wrapper.

The **Foreign Data Wrapper** dialog organizes the development of a foreign data wrapper through the following dialog tabs: **General**, **Definition**, **Options**, and **Security**. The **SQL** tab displays the SQL code generated by dialog selections.

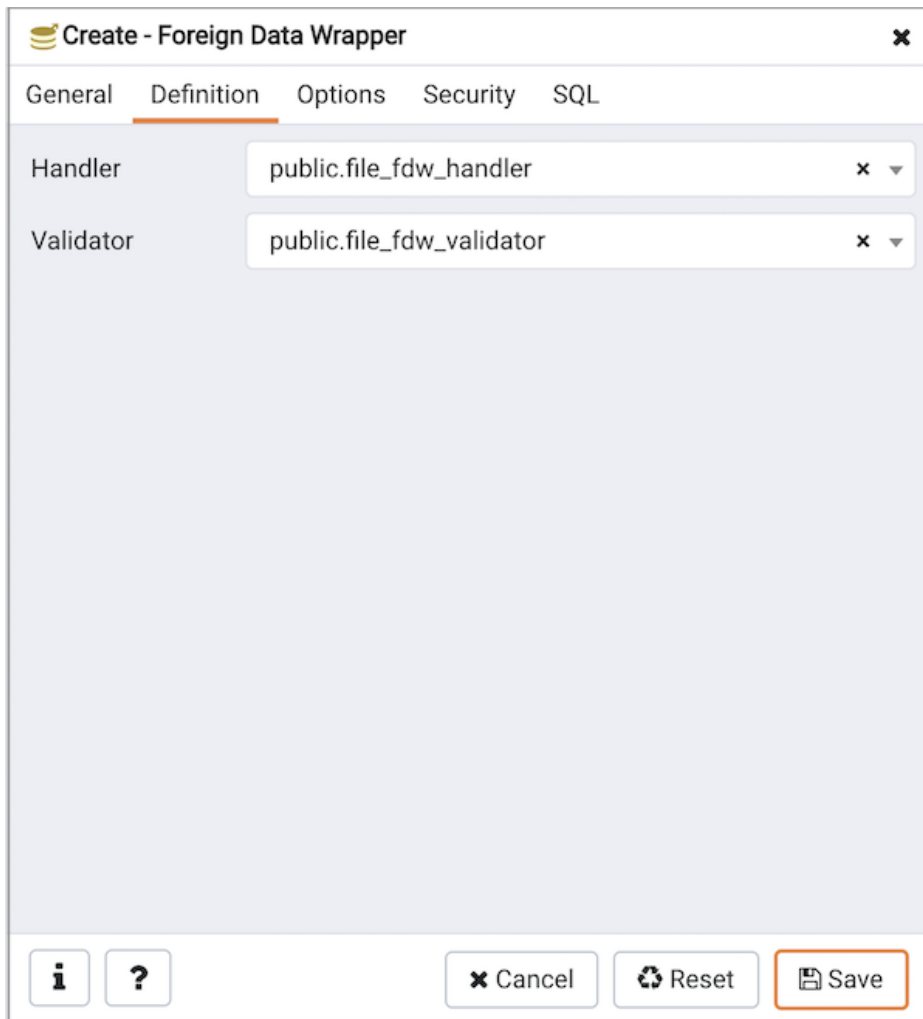


The image shows a 'Create - Foreign Data Wrapper' dialog box with a close button (X) in the top right corner. It has five tabs: 'General' (selected), 'Definition', 'Options', 'Security', and 'SQL'. The 'General' tab contains three fields: 'Name' with the value 'lib\_qp\_debug', 'Owner' with a dropdown menu showing 'postgres' and a user icon, and 'Comment' with the text 'This FDW enables debugging'. At the bottom, there are four buttons: an information icon (i), a question mark icon (?), a 'Cancel' button, a 'Reset' button, and a 'Save' button which is highlighted with an orange border.

Use the fields in the **General** tab to identify the foreign data wrapper:

- Use the **Name** field to add a descriptive name for the foreign data wrapper. A foreign data wrapper name must be unique within the database. The name will be displayed in the **Browser** tree control.
- Use the drop-down listbox next to **Owner** to select the name of the role that will own the foreign data wrapper.
- Store notes about the foreign data wrapper in the **Comment** field.

Click the **Definition** tab to continue.



The image shows a dialog box titled "Create - Foreign Data Wrapper" with a close button (X) in the top right corner. The dialog has five tabs: "General", "Definition", "Options", "Security", and "SQL". The "Definition" tab is currently selected and highlighted with an orange underline. Inside the "Definition" tab, there are two fields: "Handler" and "Validator". The "Handler" field contains the text "public.file\_fdw\_handler" and has a small "x" icon and a dropdown arrow to its right. The "Validator" field contains the text "public.file\_fdw\_validator" and also has a small "x" icon and a dropdown arrow to its right. At the bottom of the dialog, there are four buttons: an information icon (i), a question mark icon (?), a "Cancel" button with an "x" icon, a "Reset" button with a circular arrow icon, and a "Save" button with a floppy disk icon. The "Save" button is highlighted with an orange border.

Use the fields in the **Definition** tab to set parameters:

- Select the name of the handler function from the drop-down listbox in the **Handler** field. This is the name of an existing function that will be called to retrieve the execution functions for foreign tables.
- Select the name of the validator function from the drop-down listbox in the **Validator** field. This is the name of an existing function that will be called to check the generic options given to the foreign data wrapper, as well as options for foreign servers, user mappings and foreign tables using the foreign data wrapper.

Click the **Options** tab to continue.

**Create - Foreign Data Wrapper** [X]

General Definition **Options** Security SQL

Options [ + ]

| Option | Value |
|--------|-------|
|--------|-------|

[ i ] [ ? ] [ X Cancel ] [ Reset ] [ Save ]

Use the fields in the **Options** tab to specify options:

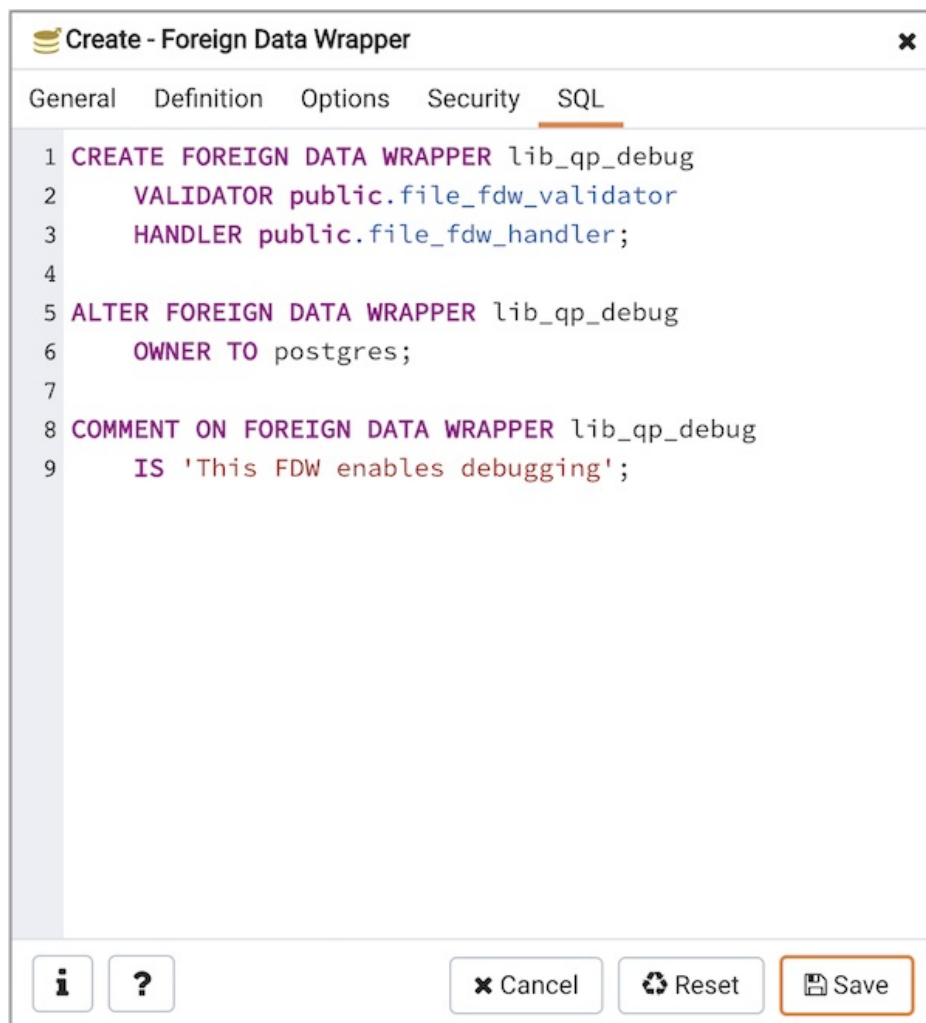
- Click the **Add** icon (+) button to add an option/value pair for the foreign data wrapper. Supported option/value pairs will be specific to the selected foreign data wrapper.
- Specify the option name in the **Option** field and provide a corresponding value in the **Value** field.

Click the **Add** icon (+) to specify each additional pair; to discard an option, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Click the **Security** tab to continue.







The example creates a foreign data wrapper named `libqp_debug` that uses pre-existing validator and handler functions, `dblink_fdw_validator` and `libpg_fdw_handler`. Selections on the `Options` tab set `debug` equal to `true`. The foreign data wrapper is owned by `postgres`.

- Click the `Help` button (?) to access online help. View context-sensitive help in the `Tabbed browser`, where a new tab displays the PostgreSQL core documentation.
- Click the `Save` button to save work.
- Click the `Cancel` button to exit without saving work.
- Click the `Reset` button to restore configuration parameters.

### 34.5.11.8 Foreign Server Dialog

Use the `Foreign Server` dialog to create a foreign server. A foreign server typically encapsulates connection information that a foreign-data wrapper uses to access an external data resource. Each foreign data wrapper may connect to a different foreign server; in the `Browser` tree control, expand the node of the applicable foreign data wrapper to launch the `Foreign Server` dialog.

The `Foreign Server` dialog organizes the development of a foreign server through the following dialog tabs: `General`, `Definition`, `Options`, and `Security`. The `SQL` tab displays the SQL code generated by dialog selections.

**Create - Foreign Server** [X]

General Definition Options Security SQL

Name:

Owner: postgres ▼

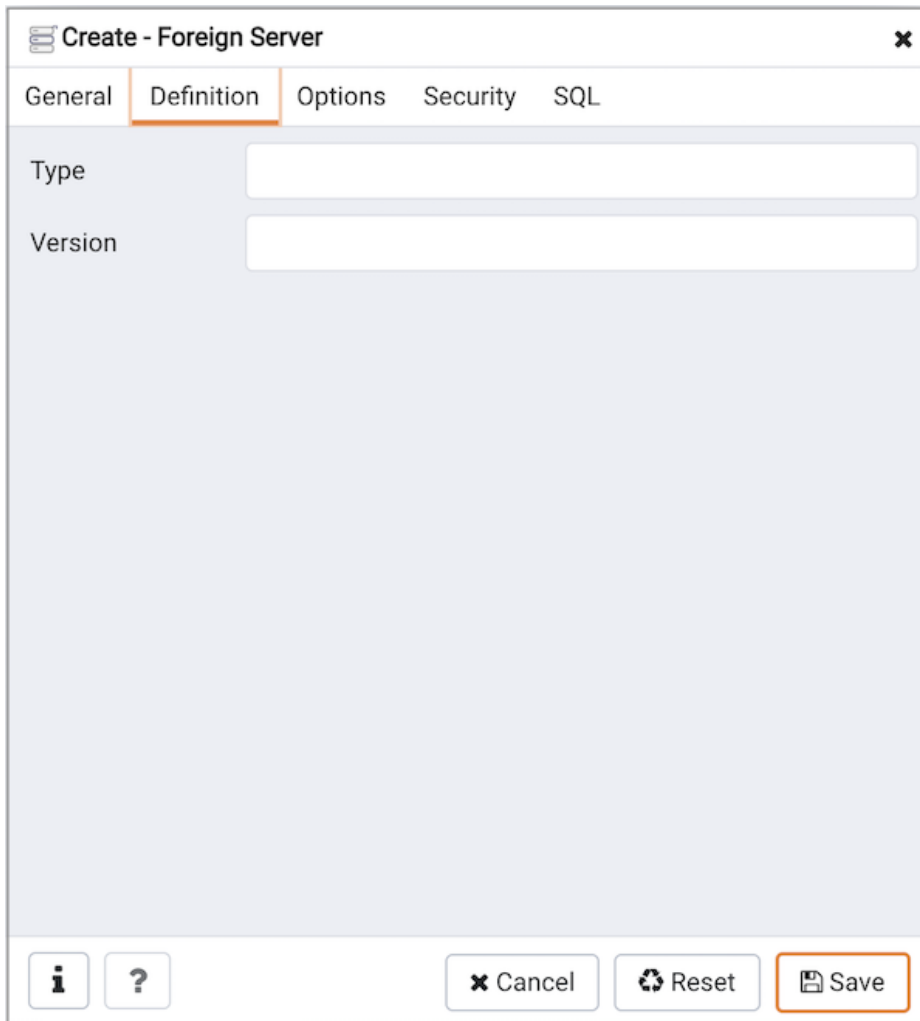
Comment:

Cancel Reset Save

Use the fields in the **General** tab to identify the foreign server:

- Use the **Name** field to add a descriptive name for the foreign server. The name will be displayed in the **Browser** tree control. It must be unique within the database.
- Use the drop-down listbox next to **Owner** to select a role.
- Store notes about the foreign server in the **Comment** field.

Click the **Definition** tab to continue.



The image shows a 'Create - Foreign Server' dialog box with a close button (X) in the top right corner. It has five tabs: 'General', 'Definition', 'Options', 'Security', and 'SQL'. The 'Definition' tab is selected and highlighted with an orange border. Inside the 'Definition' tab, there are two input fields: 'Type' and 'Version'. The 'Type' field is a text box, and the 'Version' field is a text box. At the bottom of the dialog, there are five buttons: an information icon (i), a help icon (?), a 'Cancel' button with an X icon, a 'Reset' button with a circular arrow icon, and a 'Save' button with a floppy disk icon. The 'Save' button is highlighted with an orange border.

Use the fields in the **Definition** tab to set parameters:

- Use the **Type** field to specify a server type.
- Use the **Version** field to specify a server version.

Click the **Options** tab to continue.

Create - Foreign Server

General

Definition

Options

Security

SQL

Options

|  | Options | Value     |
|--|---------|-----------|
|  | host    | 127.0.0.1 |
|  | port    | 5432      |
|  | dbname  | db01      |

i

?

Cancel

Reset

Save

Use the fields in the **Options** tab to specify options. Click the **Add** button to create an option clause for the foreign server.

- Specify the option name in the **Option** field.
- Provide a corresponding value in the **Value** field.

Click **Add** to create each additional clause; to discard an option, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Click the **Security** tab to continue.

Create - Foreign Server

General

Definition

Options

Security

SQL

Privileges

| Grantee | Privileges | Grantor |
|---------|------------|---------|
|---------|------------|---------|

i

?

Cancel

Reset

Save

Use the **Security** tab to assign security privileges to the foreign server. Click **Add** before you assign a set of privileges.

- Select the name of the role from the drop-down listbox in the **Grantee** field.
- Click inside the **Privileges** field. Check the boxes to the left of one or more privileges to grant the selected privileges to the specified user.
- The current user, who is the default grantor for granting the privilege, is displayed in the **Grantor** field.

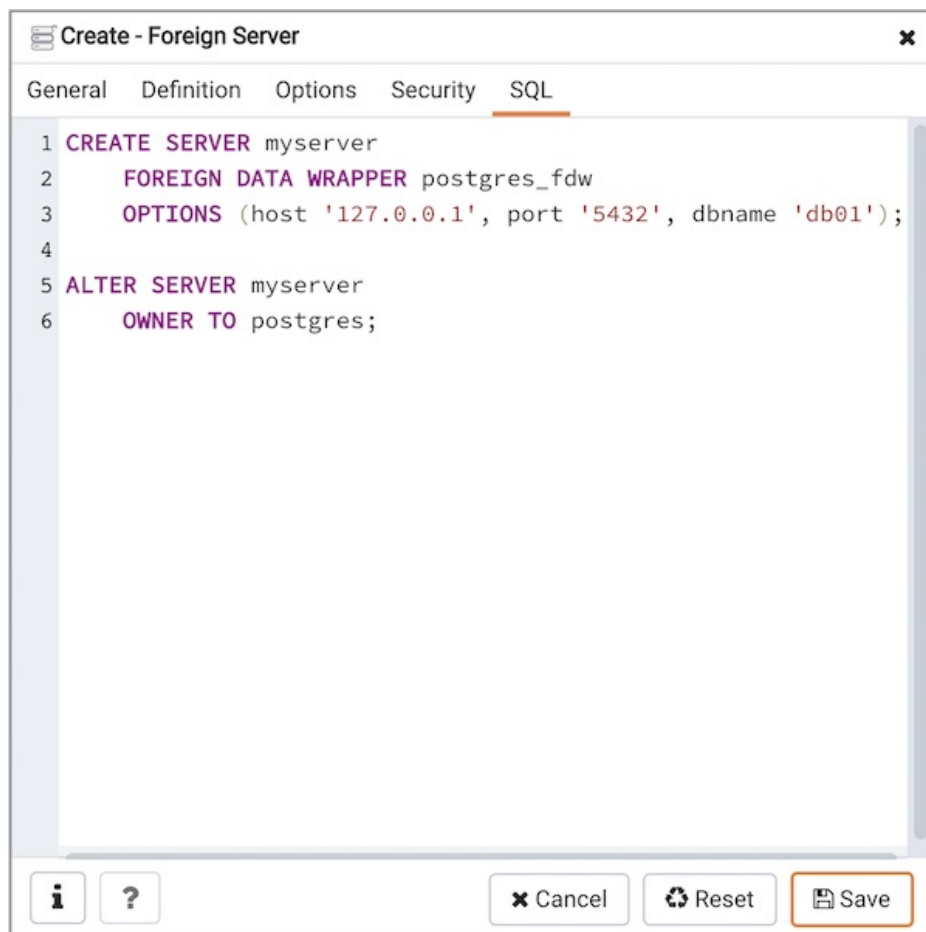
Click **Add** to assign a new set of privileges; to discard a privilege, click the trash icon to the left of the row and confirm deletion in the **Delete Row** dialog.

Click the **SQL** tab to continue.

Your entries in the **Foreign Server** dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

Example

The following is an example of the sql command generated by user selections in the **Foreign Server** dialog:



The example shown demonstrates creating a foreign server for the foreign data wrapper `hdfs_fdw`. It has the name `hdfs_server`; its type is `hiveserver2`. Options for the foreign server include a host and a port.

- Click the **Info** button (i) to access online help. View context-sensitive help in the **Tabbed browser**, where a new tab displays the PostgreSQL core documentation.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

### 34.5.11.9 Foreign Table Dialog

Use the **Foreign Table** dialog to define a foreign table in the current database. Foreign tables define the structure of an external data source that resides on a foreign server.

The **Foreign Table** dialog organizes the development of a foreign table through the following dialog tabs: **General**, **Definition**, **Columns**, **Constraints**, **Options**, and **Security**. The **SQL** tab displays the SQL code generated by dialog selections.

Create - Foreign Table

General

Definition

Columns

Constraints

Options

Security

SQL

Name

test

Owner

postgres

Schema

public

Comment

i

?

Cancel

Reset

Save

Use the fields in the **General** tab to identify the foreign table:

- Use the **Name** field to add a descriptive name for the foreign table. The name of the foreign table must be distinct from the name of any other foreign table, table, sequence, index, view, existing data type, or materialized view in the same schema. The name will be displayed in the **Browser** tree control.
- Use the drop-down listbox next to **Owner** to select the name of the role that will own the foreign table.
- Select the name of the schema in which the foreign table will reside from the drop-down listbox in the **Schema** field.
- Store notes about the foreign table in the **Comment** field.

Click the **Definition** tab to continue.



Create - Foreign Table

General

Definition

Columns

Constraints

Options

Security

SQL

Foreign server

myserver

x

Inherits

x public.try

x

i

?

x Cancel

Reset

Save

Use the fields in the **Definition** tab to define the external data source:

- Use the drop-down listbox next to **Foreign server** to select a foreign server. This list is populated with servers defined through the **Foreign Server** dialog.
- Use the drop-down listbox next to **Inherits** to specify a parent table. The foreign table will inherit all of its columns. This field is optional.

Click the **Columns** tab to continue.

Create - Foreign Table

General

Definition

Columns

Constraints

Options

Security

SQL

Columns

+

|                         | Name | Data type |
|-------------------------|------|-----------|
| <div></div> <div></div> | No   | numeric   |

i

?

Cancel

Reset

Save

Use the fields in the **Columns** tab to add columns and their attributes to the table. Click the **Add** icon (+) to define a column:

- Use the **Name** field to add a descriptive name for the column.
- Use the drop-down listbox in the **Data Type** field to select a data type for the column. This can include array specifiers. For more information on which data types are supported by PostgreSQL, refer to Chapter 8 of the core documentation.

Click the **Add** icon (+) to specify each additional column; to discard a column, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Click the **Constraints** tab to continue.

Create - Foreign Table

General

Definition

Columns

Constraints

Options

Security

SQL

Constraints

| Name | Check | No inherit? | Validate? |
|------|-------|-------------|-----------|
|------|-------|-------------|-----------|

i

?

Cancel

Reset

Save

Use the fields in the **Constraints** tab to apply a table constraint to the foreign table. Click the **Add** icon (+) to define a constraint:

- Use the **Name** field to add a descriptive name for the constraint. If the constraint is violated, the constraint name is present in error messages, so constraint names like **col must be positive** can be used to communicate helpful information.
- Use the **Check** field to write a check expression producing a Boolean result. Each row in the foreign table is expected to satisfy the check expression.
- Check the **No Inherit** checkbox to specify that the constraint will not propagate to child tables.
- Uncheck the **Validate** checkbox to disable validation. The database will not assume that the constraint holds for all rows in the table.

Click the **Add** icon (+) to specify each additional constraint; to discard a constraint, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Click the **Options** tab to continue.

Create - Foreign Table

General

Definition

Columns

Constraints

Options

Security

SQL

Options

Option

Value

i

?

Cancel

Reset

Save

Use the fields in the **Options** tab to specify options to be associated with the new foreign table or one of its columns; the accepted option names and values are specific to the foreign data wrapper associated with the foreign server. Click the **Add** icon (+) to add an option/value pair.

- Specify the option name in the **Option** field. Duplicate option names are not allowed.
- Provide a corresponding value in the **Value** field.

Click the **Add** icon (+) to specify each additional option/value pair; to discard an option, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Click the **Security** tab to continue.

Create - Foreign Table

General

Definition

Columns

Constraints

Options

Security

SQL

Privileges

Grantee

Privileges

Grantor

postgres

ALL

WITH GRANT OPTION

INSERT

WITH GRANT OPTION

SELECT

WITH GRANT OPTION

UPDATE

WITH GRANT OPTION

REFERENCES

WITH GRANT OPTION

postgres

Security labels

Provider

Security label

i

?

Cancel

Reset

Save

Use the **Security** tab to assign privileges and define security labels.

Use the **Privileges** panel to assign privileges to a role. Click the **Add** icon (+) to set privileges for database objects:

- Select the name of the role to which privileges will be assigned from the drop-down listbox in the **Grantee** field.
- Click inside the **Privileges** field. Check the boxes to the left of one or more privileges to grant the selected privilege to the specified user.
- The current user, who is the default grantor for granting the privilege, is displayed in the **Grantor** field.

Click the **Add** icon (+) to assign additional privileges; to discard a privilege, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Use the **Security Labels** panel to define security labels applied to the function. Click the **Add** icon (+) to add each security label selection:

- Specify a security label provider in the **Provider** field. The named provider must be loaded and must consent to the proposed labeling operation.
- Specify a security label in the **Security Label** field. The meaning of a given label is at the discretion of the label provider. PostgreSQL places no restrictions on whether or how a label provider must interpret security labels; it merely provides a mechanism for storing them.

Click the **Add** icon (+) to assign additional security labels; to discard a security label, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Click the **SQL** tab to continue.

Your entries in the **Foreign Table** dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

Example

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

687

The following is an example of the sql command generated by user selections in the **Foreign Table** dialog:



The example shown demonstrates creating a foreign table **weblogs** with multiple columns and two options.

- Click the **Info** button (i) to access online help. View context-sensitive help in the **Tabbed browser**, where a new tab displays the PostgreSQL core documentation.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

### 34.5.11.10 FTS Configuration dialog

Use the **FTS Configuration** dialog to configure a full text search. A text search configuration specifies a text search parser that can divide a string into tokens, along with dictionaries that can identify searchable tokens.

The **FTS Configuration** dialog organizes the development of a FTS configuration through the following dialog tabs: "**General**", **Definition**, and **Tokens**. The **SQL** tab displays the SQL code generated by dialog selections.

Click the **General** tab to begin.

**Create - FTS Configuration**

General Definition Tokens SQL

Name: meme\_phrases

Owner: postgres

Schema: public

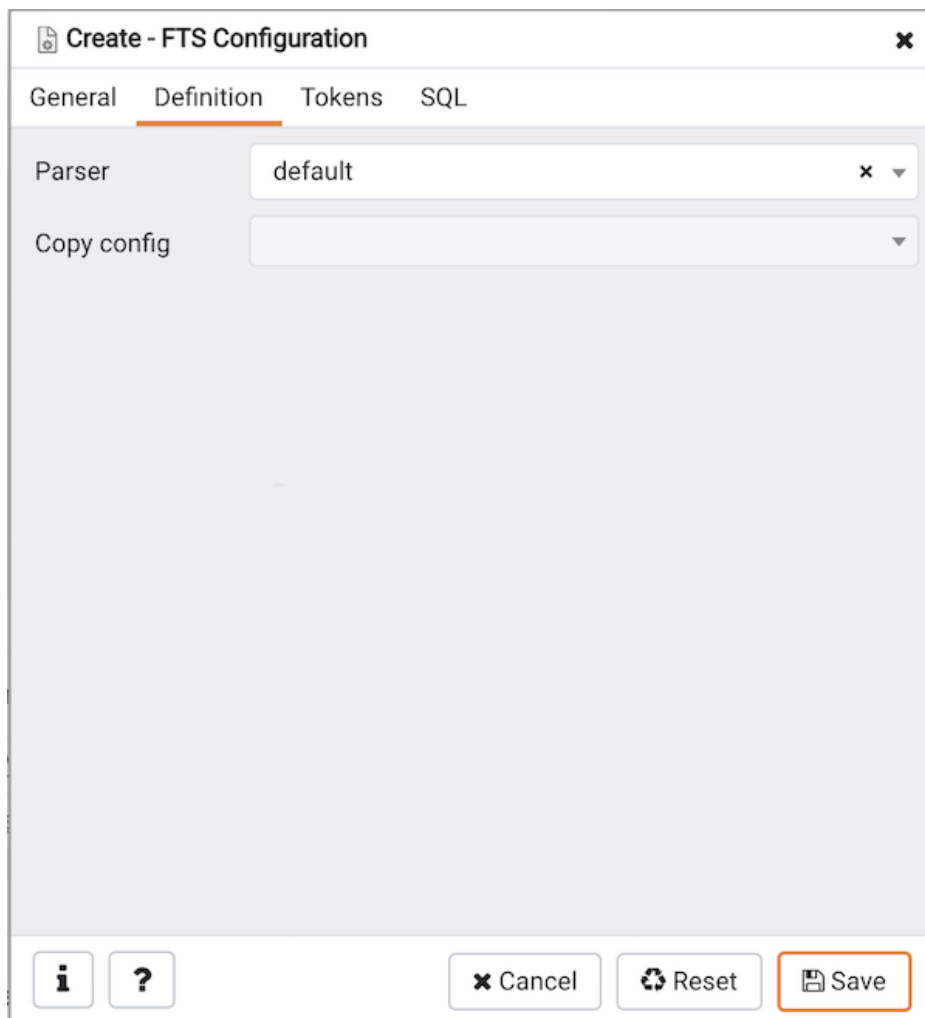
Comment:

Buttons: [i] [?] [x Cancel] [Reset] [Save]

Use the fields in the **General** tab to identify a FTS configuration:

- Use the **Name** field to add a descriptive name for the FTS configuration. The name will be displayed in the **Browser** tree control.
- Use the drop-down listbox next to **Owner** to specify the role that will own the configuration.
- Select the name of the schema in which the FTS configuration will reside from the drop-down listbox in the **Schema** field.
- Store notes about the FTS configuration in the **Comment** field.

Click the **Definition** tab to continue.



The image shows a dialog box titled "Create - FTS Configuration" with a close button (X) in the top right corner. The dialog has four tabs: "General", "Definition", "Tokens", and "SQL". The "Definition" tab is currently selected and highlighted with an orange underline. Inside the "Definition" tab, there are two fields: "Parser" and "Copy config". The "Parser" field is a text input with the value "default" and a small "x" icon to its right. The "Copy config" field is a text input that is currently empty. At the bottom of the dialog, there are four buttons: an information icon (i), a question mark icon (?), a "Cancel" button with an "x" icon, a "Reset" button with a circular arrow icon, and a "Save" button with a floppy disk icon. The "Save" button is highlighted with an orange border.

Use the fields in the **Definition** tab to define parameters:

- Select the name of the text search parser from the drop-down listbox in the **Parser** field.
- Select a language from the drop-down listbox in the **Copy Config** field.

Click the **Tokens** tab to continue.



Create - FTS Configuration

General

Definition

Tokens

SQL

Tokens

+

| Token | Dictionaries |
|-------|--------------|
|-------|--------------|

i

?

Cancel

Reset

Save

Use the fields in the **Tokens** tab to add a token:

- Use the **Tokens** field to specify the name of a token.
- Click the **Add** icon (+) to create a token.
- Use the **Dictionaries** field to specify a dictionary.

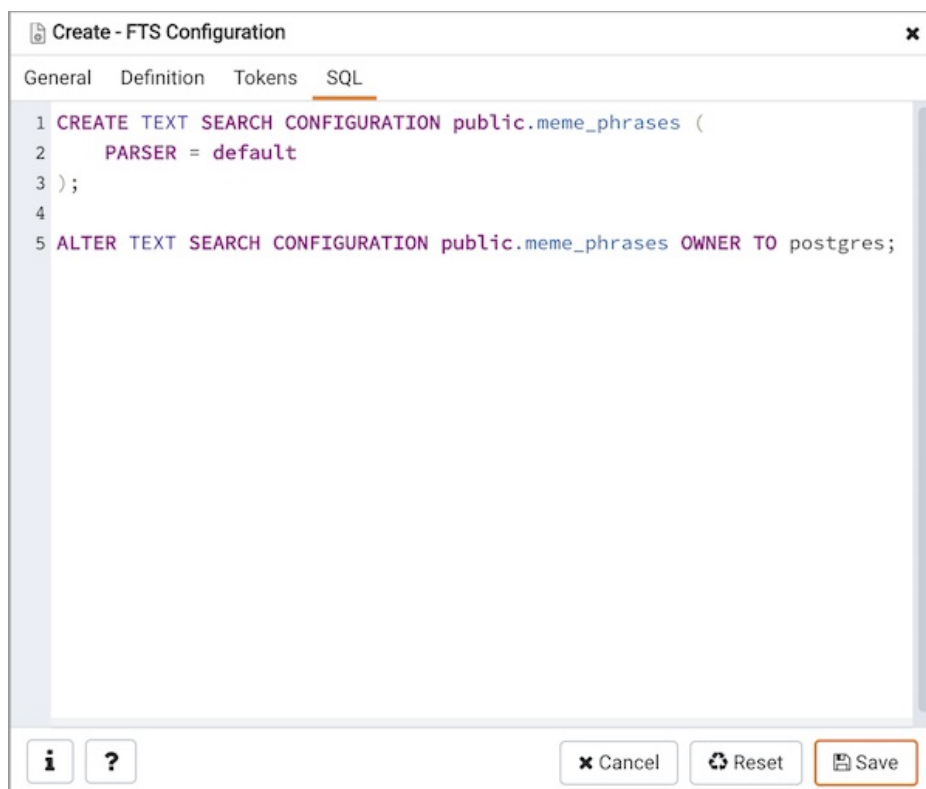
Repeat these steps to add additional tokens; to discard a token, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Click the **SQL** tab to continue.

Your entries in the **FTS Configuration** dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

Example

The following is an example of the sql command generated by user selections in the **FTS Configuration** dialog:



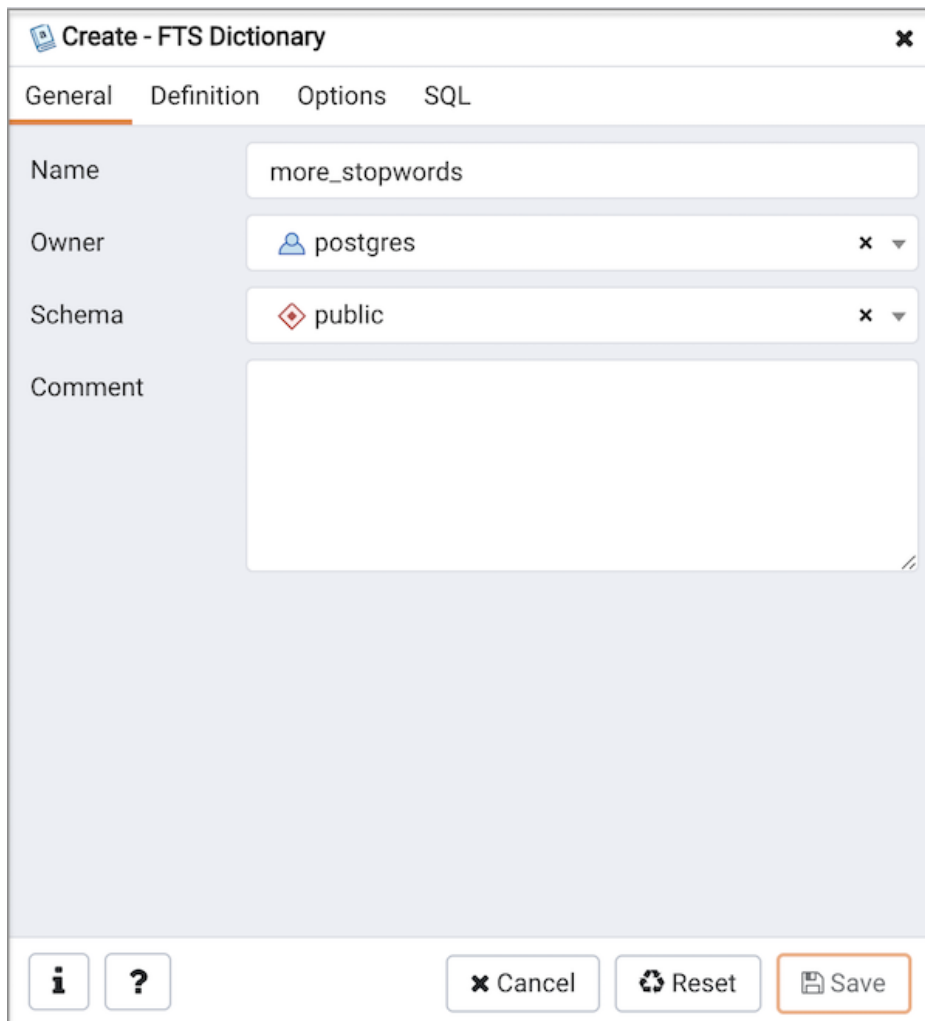
The example shown demonstrates creating a FTS configuration named `meme_phrases`. It uses the `default` parser.

- Click the **Info** button (i) to access online help. View context-sensitive help in the **Tabbed browser**, where a new tab displays the PostgreSQL core documentation.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

### 34.5.11.11 FTS Dictionary Dialog

Use the **FTS Dictionary** dialog to create a full text search dictionary. You can use a predefined templates or create a new dictionary with custom parameters.

The **FTS Dictionary** dialog organizes the development of a FTS dictionary through the following dialog tabs: **General**, **Definition**, and **Options**. The **SQL** tab displays the SQL code generated by dialog selections.



The image shows a 'Create - FTS Dictionary' dialog box with four tabs: General, Definition, Options, and SQL. The 'General' tab is selected. It contains the following fields:

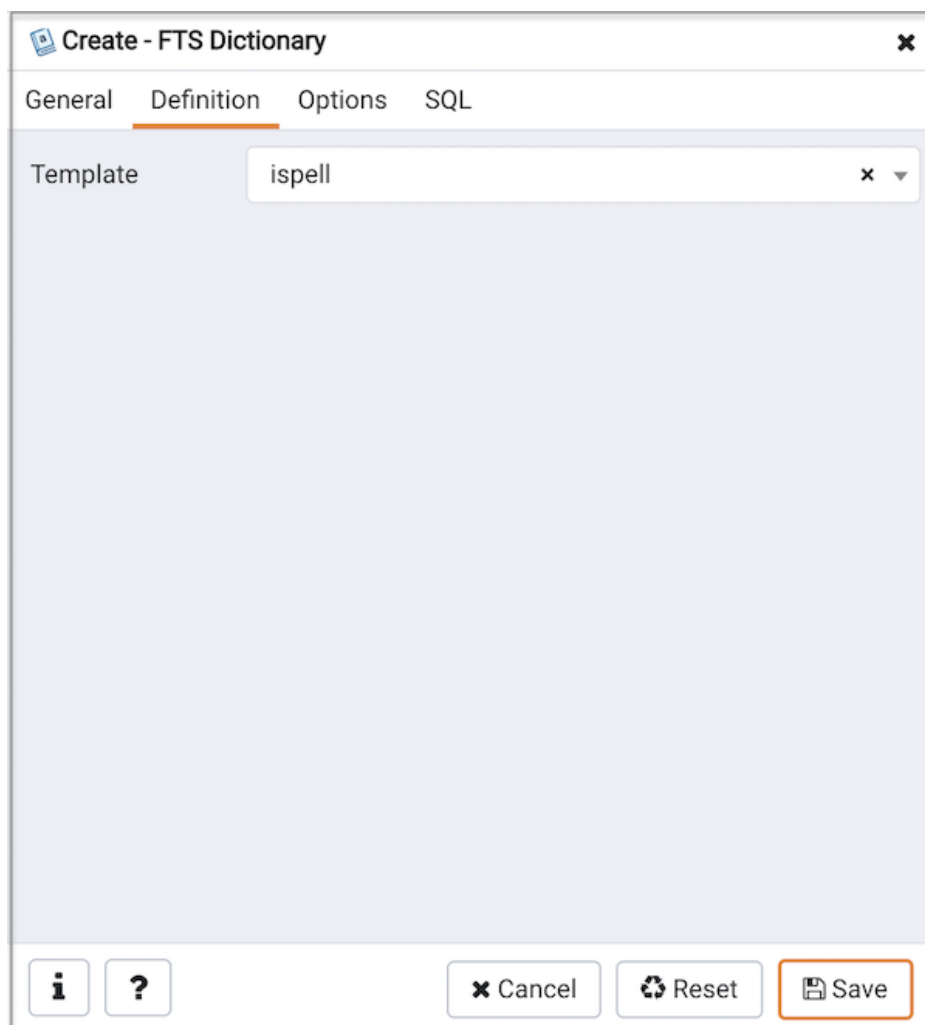
- Name:** A text field containing 'more\_stopwords'.
- Owner:** A dropdown menu showing 'postgres' with a user icon and a close button.
- Schema:** A dropdown menu showing 'public' with a schema icon and a close button.
- Comment:** A large text area for notes.

At the bottom, there are buttons for 'i' (info), '?' (help), 'x Cancel', 'Reset' (with a refresh icon), and 'Save' (with a save icon).

Use the fields in the **General** tab to identify the dictionary:

- Use the **Name** field to add a descriptive name for the dictionary. The name will be displayed in the **Browser** tree control.
- Use the drop-down listbox next to **Owner** to select the role that will own the FTS Dictionary.
- Select the name of the schema in which the dictionary will reside from the drop-down listbox in the **Schema** field.
- Store notes about the dictionary in the **Comment** field.

Click the **Definition** tab to continue.



Use the field in the **Definition** tab to choose a template from the drop-down listbox:

- Select **ispell** to select the Ispell template. The Ispell dictionary template supports morphological dictionaries, which can normalize many different linguistic forms of a word into the same lexeme. For example, an English Ispell dictionary can match all declensions and conjugations of the search term bank, e.g., banking, banked, banks, banks', and bank's. Ispell dictionaries usually recognize a limited set of words, so they should be followed by another broader dictionary; for example, a Snowball dictionary, which recognizes everything.
- Select **simple** to select the simple template. The simple dictionary template operates by converting the input token to lower case and checking it against a file of stop words. If it is found in the file then an empty array is returned, causing the token to be discarded. If not, the lower-cased form of the word is returned as the normalized lexeme. Alternatively, the dictionary can be configured to report non-stop-words as unrecognized, allowing them to be passed on to the next dictionary in the list.
- Select **snowball** to select the Snowball template. The Snowball dictionary template is based on a project by Martin Porter, inventor of the popular Porter's stemming algorithm for the English language. Snowball now provides stemming algorithms for many languages (see the Snowball site for more information). Each algorithm understands how to reduce common variant forms of words to a base, or stem, spelling within its language. A Snowball dictionary recognizes everything, whether or not it is able to simplify the word, so it should be placed at the end of the dictionary list. It is useless to have it before any other dictionary because a token will never pass through it to the next dictionary.
- Select **synonym** to select the synonym template. This dictionary template is used to create dictionaries that replace a word with a synonym. Phrases are not supported (use the thesaurus template (Section 12.6.4) for that). A synonym dictionary can be used to overcome linguistic problems, for example, to prevent an English stemmer dictionary from reducing the word Paris to pari.
- Select **thesaurus** to select the thesaurus template. A thesaurus dictionary replaces all non-preferred terms by one preferred term and, optionally, preserves the original terms for indexing as well. PostgreSQL's current implementation of the thesaurus dictionary is an extension of the synonym dictionary with added phrase support.

Click the **Options** tab to continue.

Create - FTS Dictionary

General

Definition

Options

SQL

Option

+

|  | Option      | Value      |
|--|-------------|------------|
|  | data_option | data_value |

i

?

✕ Cancel

Reset

Save

Use the fields in the **Options** tab to provide template-specific options. Click the **Add** icon (+) to add an option clause:

- Specify the name of an option in the **Option** field
- Provide a value for the option in the **Value** field.

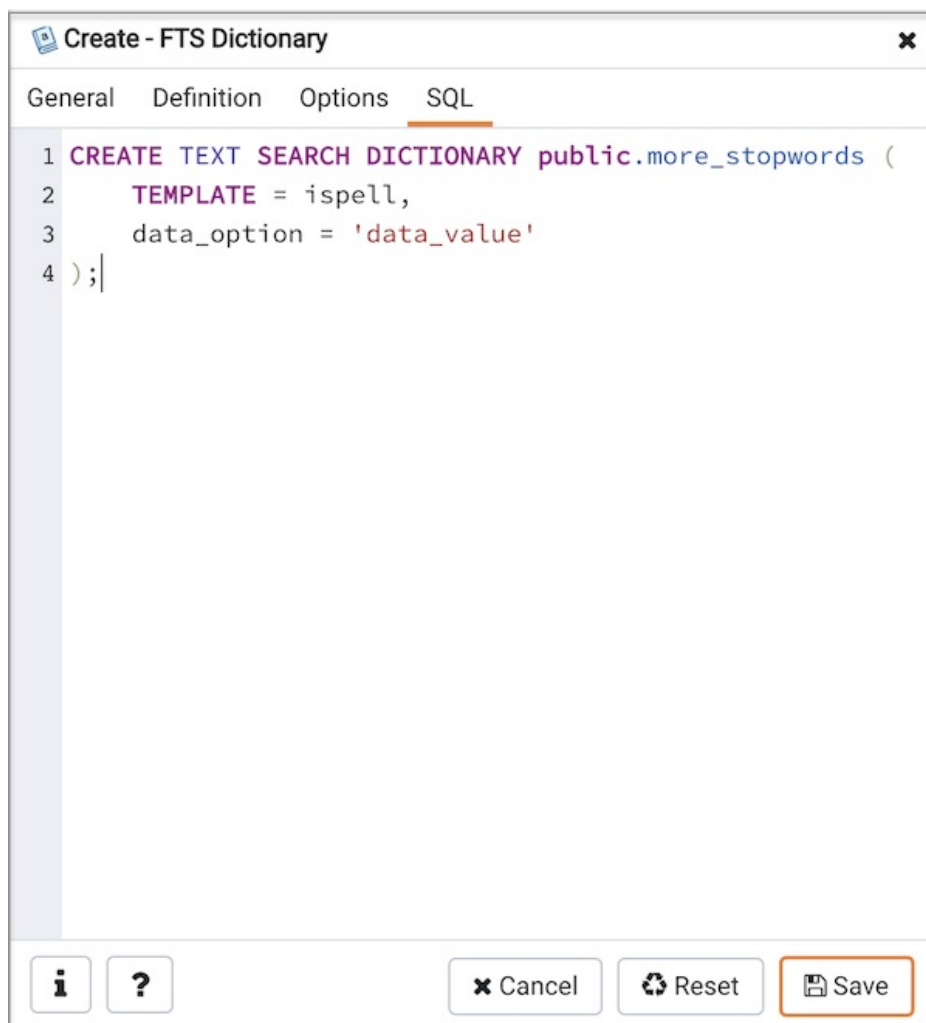
Click the **Add** icon (+) to specify each additional option/value pair; to discard an option, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Click the **SQL** tab to continue.

Your entries in the **FTS Dictionary** dialog generate a generate a SQL command. Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

Example

The following is an example of the sql command generated by user selections in the **FTS Dictionary** dialog:



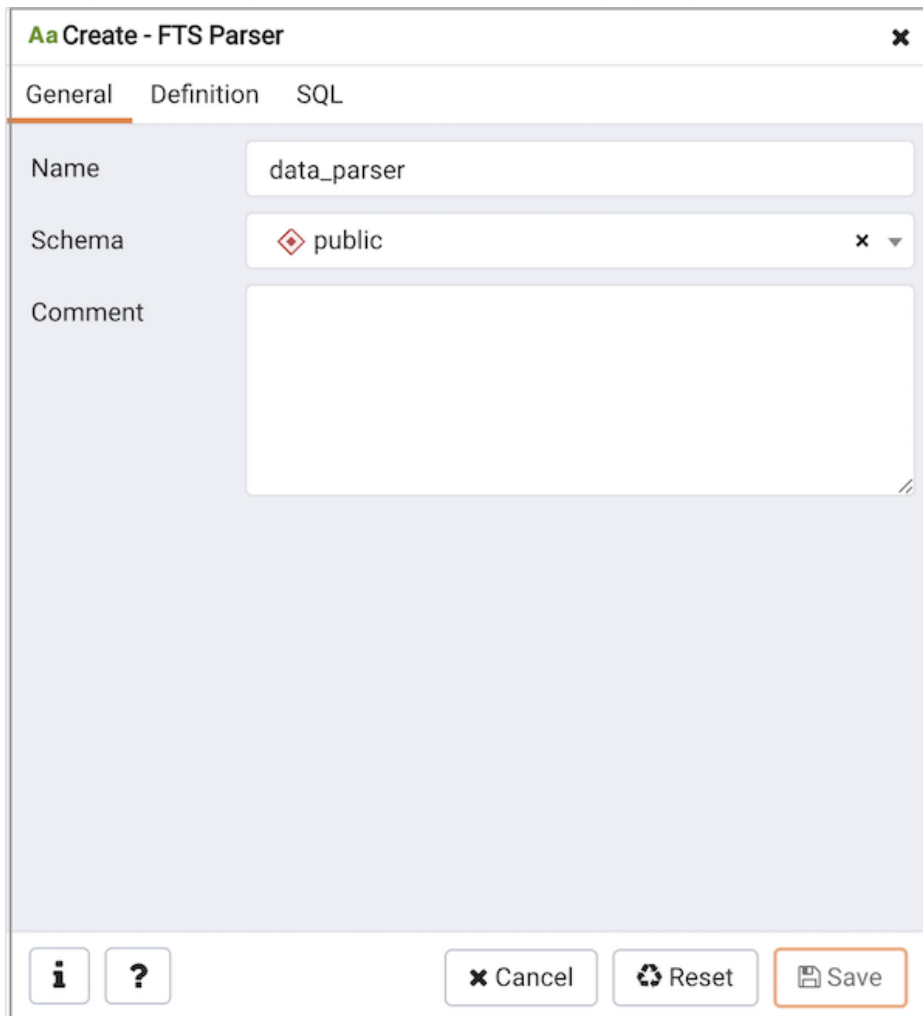
The example shown demonstrates creating a custom dictionary named `more_stopwords` which is based on the simple template and is configured to use standard English.

- Click the **Info** button (i) to access online help. View context-sensitive help in the **Tabbed browser**, where a new tab displays the PostgreSQL core documentation.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

### 34.5.11.12 FTS Parser Dialog

Use the **FTS Parser** dialog to create a new text search parser. A text search parser defines a method for splitting a text string into tokens and assigning types (categories) to the tokens.

The **FTS Parser** dialog organizes the development of a text search parser through the following dialog tabs: **General**, and **Definition**. The **SQL** tab displays the SQL code generated by dialog selections.



**Create - FTS Parser**

General Definition SQL

Name: data\_parser

Schema: public

Comment:

Buttons: [i] [?] [x] Cancel [Reset] [Save]

Use the fields in the **General** tab to identify a text search parser:

- Use the **Name** field to add a descriptive name for the parser. The name will be displayed in the **Browser** tree control.
- Select the name of the schema in which the parser will reside from the drop-down listbox in the **Schema** field.
- Store notes about the domain in the **Comment** field.

Click the **Definition** tab to continue.

Aa Create - FTS Parser

×

General

Definition

SQL

Start function

int4\_accum

×

▼

Get next token function

prsd\_nexttoken

×

▼

End function

bpchar\_sortsupport

×

▼

Lextypes function

brin\_inclusion\_opcinfo

×

▼

Headline function

prsd\_headline

×

▼

i

?

×

Cancel

↺

Reset

💾

Save

Use the fields in the **Definition** tab to define parameters:

- Use the drop-down listbox next to **Start function** to select the name of the function that will initialize the parser.
- Use the drop-down listbox next to **Get next token function** to select the name of the function that will return the next token.
- Use the drop-down listbox next to **End function** to select the name of the function that is called when the parser is finished.
- Use the drop-down listbox next to **Lextypes function** to select the name of the lextypes function for the parser. The lextypes function returns an array that contains the id, alias, and a description of the tokens used by the parser.
- Use the drop-down listbox next to **Headline function** to select the name of the headline function for the parser. The headline function returns an excerpt from the document in which the terms of the query are highlighted.

Click the **SQL** tab to continue.





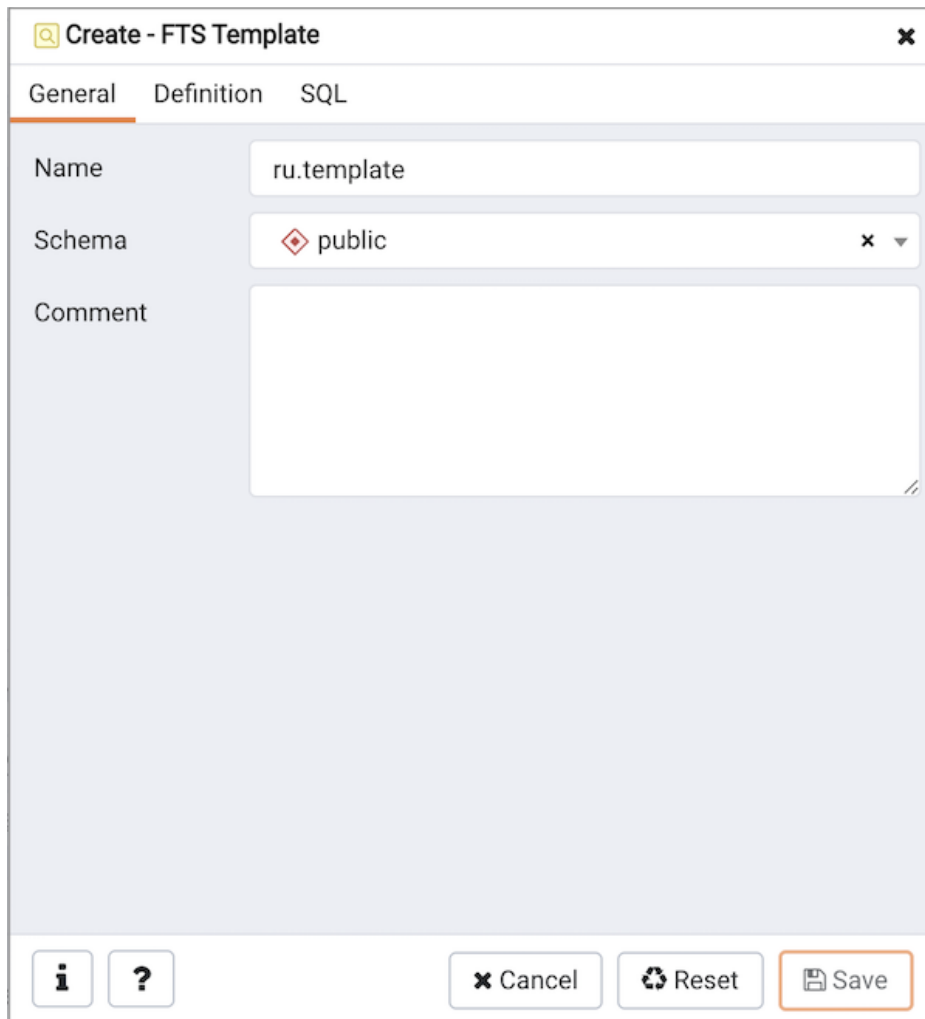
Your entries in the **FTS Parser** dialog generate a generate a SQL command. Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

- Click the **Info** button (i) to access online help. View context-sensitive help in the **Tabbed browser**, where a new tab displays the PostgreSQL core documentation.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

### 34.5.11.13 FTS Template Dialog

Use the **FTS Template** dialog to create a new text search template. A text search template defines the functions that implement text search dictionaries.

The **FTS Template** dialog organizes the development of a text search Template through the following dialog tabs: **General**, and **Definition**. The **SQL** tab displays the SQL code generated by dialog selections.



The image shows a 'Create - FTS Template' dialog box with three tabs: 'General', 'Definition', and 'SQL'. The 'General' tab is active. It contains three fields: 'Name' with the value 'ru.template', 'Schema' with a dropdown menu showing 'public', and a large 'Comment' text area. At the bottom, there are buttons for 'Cancel', 'Reset', and 'Save' (which is highlighted with an orange border). There are also information and help icons on the left.

Create - FTS Template

General Definition SQL

Name ru.template

Schema public

Comment

Cancel Reset Save

Use the fields in the **General** tab to identify a template:

- Use the **Name** field to add a descriptive name for the template. The name will be displayed in the **Browser** tree control.
- Select the name of the schema in which the template will reside from the drop-down listbox in the **Schema** field.
- Store notes about the template in the **Comment** field.

Click the **Definition** tab to continue.

**Create - FTS Template**

General **Definition** SQL

Init function

Lexize function

**Cancel** **Reset** **Save**

Use the fields in the **Definition** tab to define function parameters:

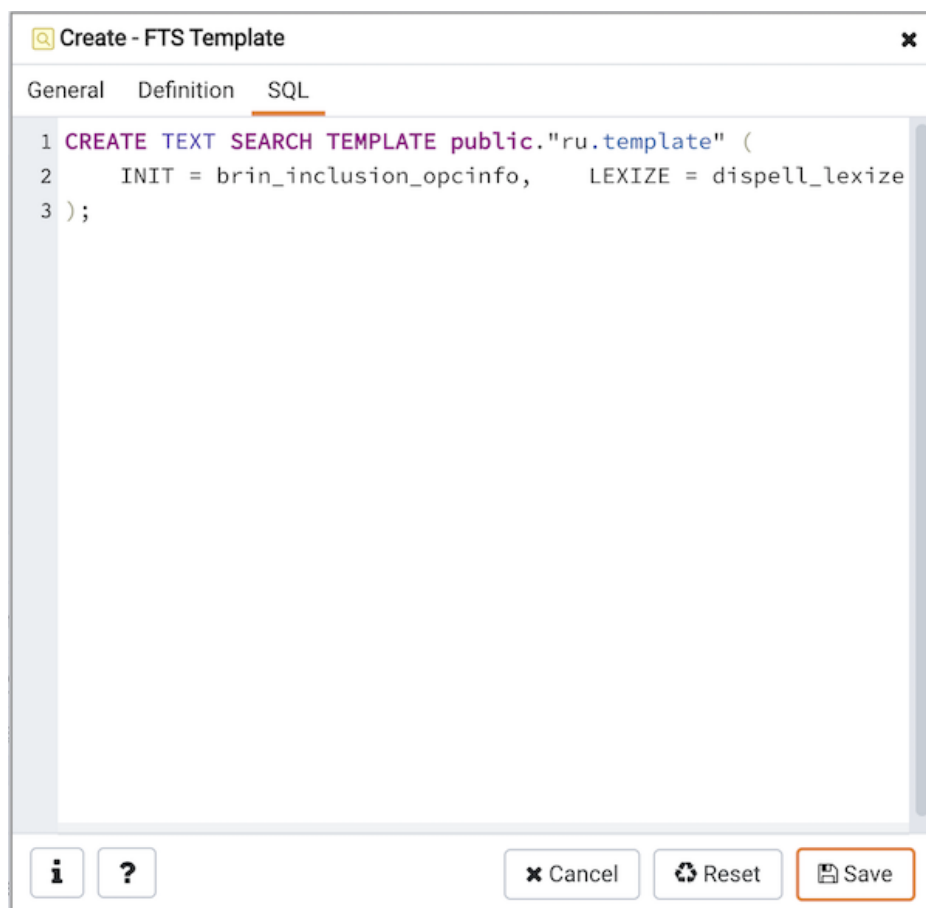
- Use the drop-down listbox next to **Init function** to select the name of the init function for the template. The init function is optional.
- Use the drop-down listbox next to **Lexize function** to select the name of the lexize function for the template. The lexize function is required.

Click the **SQL** tab to continue.

Your entries in the **FTS Template** dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

### Example

The following is an example of the sql command generated by user selections in the **FTS Template** dialog:



The example shown demonstrates creating a fts template named `ru_template` that uses the ispell dictionary.

- Click the **Info** button (i) to access online help. View context-sensitive help in the **Tabbed browser**, where a new tab displays the PostgreSQL core documentation.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

### 34.5.11.14 Function Dialog

Use the **Function** dialog to define a function. If you drop and then recreate a function, the new function is not the same entity as the old; you must drop existing rules, views, triggers, etc. that refer to the old function.

The **Function** dialog organizes the development of a function through the following dialog tabs: **General**, **Definition**, **Code**, **Options**, **Parameters**, and **Security**. The **SQL** tab displays the SQL code generated by dialog selections.

Create - Function

General

Definition

Code

Options

Parameters

Security

SQL

Name

emp\_count

Owner

postgres

Schema

public

Comment

This function gets the employee count

i

?

Cancel

Reset

Save

Use the fields in the **General** tab to identify a function:

- Use the **Name** field to add a descriptive name for the function. The name will be displayed in the **Browser** tree control.
- Use the drop-down listbox next to **Owner** to select the name of the role that will own the function.
- Use the drop-down listbox next to **Schema** to select the schema in which the function will be created.
- Store notes about the function in the **Comment** field.

Click the **Definition** tab to continue.

Create - Function

General

Definition

Code

Options

Parameters

Security

SQL

Return type

bigint

Language

sql

Arguments

| Data type | Mode | Argument name | Default |
|-----------|------|---------------|---------|
|-----------|------|---------------|---------|

i

?

Cancel

Reset

Save

Use the fields in the **Definition** tab to define the function:

- Use the drop-down listbox next to **Return type** to select the data type returned by the function, if any.

- Use the drop-down listbox next to **Language** to select the implementation language. The default is **sql**.
- Use the fields in the **Arguments** to define an argument. Click the **Add** icon (+) to set parameters and values for the argument:
  - Use the drop-down listbox in the **Data type** field to select a data type.
  - Use the drop-down listbox in the **Mode** field to select a mode. Select **IN** for an input parameter; select **OUT** for an output parameter; select **INOUT** for both an input and an output parameter; or, select **VARIADIC** to specify a VARIADIC parameter.
  - Provide a name for the argument in the **Argument Name** field.
  - Specify a default value for the argument in the **Default Value** field.

Click the **Add** icon (+) to define another argument; to discard an argument, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Click the **Code** tab to continue.

The screenshot shows a 'Create - Function' dialog box with the following components:

- Title Bar:** 'Create - Function' with a close button (X).
- Tabs:** General, Definition, **Code** (selected), Options, Parameters, Security, SQL.
- Code Area:** Contains the SQL query: `1 select count(emp_id) from employees;`
- Bottom Bar:** Includes an information icon (i), a question mark icon (?), and three buttons: 'Cancel', 'Reset', and 'Save'.

- Use the **Code** field to write the code that will execute when the function is called.

Click the **Options** tab to continue.

The screenshot shows the 'Create - Function' dialog box with the 'Options' tab selected. The 'Volatility' dropdown is set to 'STABLE'. Other options like 'Returns a set?', 'Strict?', 'Security of definer?', 'Window?', 'Leak proof?', and 'Support function' are all set to 'No'. The 'Parallel', 'Estimated cost', and 'Estimated rows' fields are empty. The 'Cancel', 'Reset', and 'Save' buttons are at the bottom right.

Use the fields in the **Options** tab to describe or modify the action of the function:

- Use the drop-down listbox next to **Volatility** to select one of the following. **VOLATILE** is the default value.
  - **VOLATILE** indicates that the function value can change even within a single table scan, so no optimizations can be made.
  - **STABLE** indicates that the function cannot modify the database, and that within a single table scan it will consistently return the same result for the same argument values.
  - **IMMUTABLE** indicates that the function cannot modify the database and always returns the same result when given the same argument values.
- Move the **Returns a Set?** switch to indicate if the function returns a set that includes multiple rows. The default is **No**.
- Move the **Strict?** switch to indicate if the function always returns NULL whenever any of its arguments are NULL. If **Yes**, the function is not executed when there are NULL arguments; instead a NULL result is assumed automatically. The default is **No**.
- Move the **Security of definer?** switch to specify that the function is to be executed with the privileges of the user that created it. The default is **No**.
- Move the **Window?** switch to indicate that the function is a window function rather than a plain function. The default is **No**. This is currently only useful for functions written in C. The WINDOW attribute cannot be changed when replacing an existing function definition. For more information about the CREATE FUNCTION command, see the PostgreSQL core documentation available at:
 

<http://www.postgresql.org/docs/current/static/functions-window.html>
- Use the **Estimated cost** field to specify a positive number representing the estimated execution cost for the function, in units of `cpu_operator_cost`. If the function returns a set, this is the cost per returned row.
- Use the **Estimated rows** field to specify a positive number giving the estimated number of rows that the query planner should expect the function to return. This is only allowed when the function is declared to return a set. The default assumption is 1000 rows.
- Move the **Leak proof?** switch to indicate whether the function has side effects. The default is **No**. This option can only be set by the superuser.
- Use the **Support function** field to specify a planner support function to use for the function.

Click the **Parameters** tab to continue.

Create - Function

General

Definition

Code

Options

Parameters

Security

SQL

+

| Name | Value |
|------|-------|
|------|-------|

i

?

Cancel

Reset

Save

Use the fields in the **Parameters** tab to specify settings that will be applied when the function is invoked. Click the **Add** icon (+) to add a **Name / Value** field in the table.

- Use the drop-down listbox in the **Name** column in the **Parameters** panel to select a parameter.
- Use the **Value** field to specify the value that will be associated with the selected variable. This field is context-sensitive.

Click the **Security** tab to continue.

Create - Function

General

Definition

Code

Options

Parameters

Security

SQL

Privileges

+

| Grantee | Privileges | Grantor |
|---------|------------|---------|
|---------|------------|---------|

Security labels

+

| Provider | Security label |
|----------|----------------|
|----------|----------------|

i

?

Cancel

Reset

Save

Use the **Security** tab to assign privileges and define security labels.



Use the **Privileges** panel to assign usage privileges for the function to a role.

- Select the name of the role from the drop-down listbox in the **Grantee** field.
- Click inside the **Privileges** field. Check the boxes to the left of one or more privileges to grant the selected privilege to the specified user.
- The current user, who is the default grantor for granting the privilege, is displayed in the **Grantor** field.

Click the **Add** icon (+) to assign additional privileges; to discard a privilege, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Use the **Security Labels** panel to define security labels applied to the function. Click the **Add** icon (+) to add each security label selection:

- Specify a security label provider in the **Provider** field. The named provider must be loaded and must consent to the proposed labeling operation.
- Specify a security label in the **Security Label** field. The meaning of a given label is at the discretion of the label provider. PostgreSQL places no restrictions on whether or how a label provider must interpret security labels; it merely provides a mechanism for storing them.

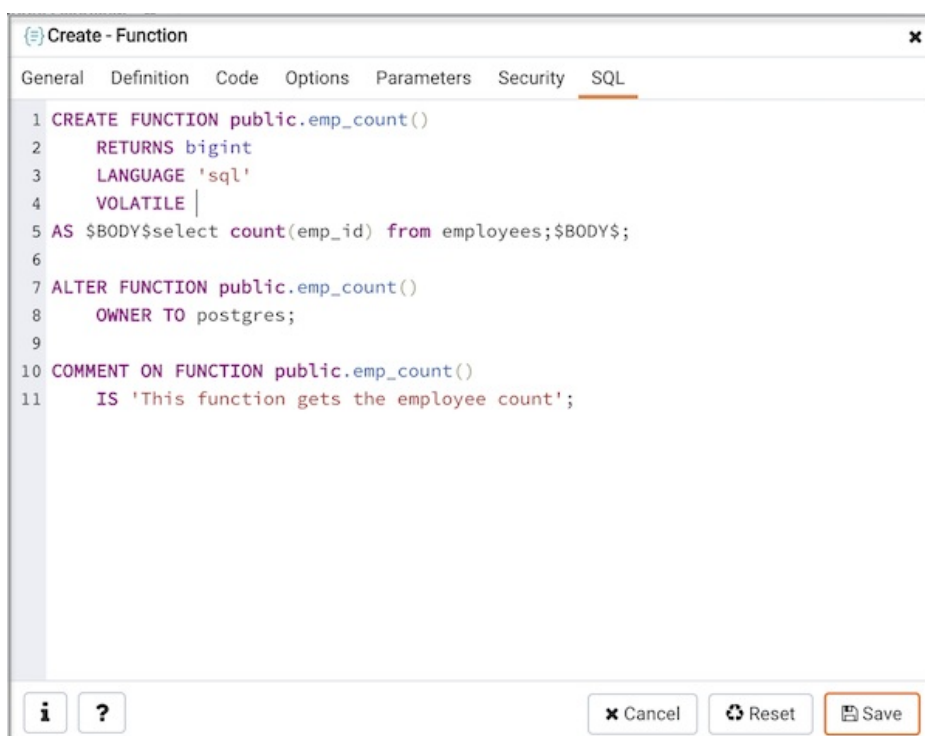
Click the **Add** icon (+) to assign additional security labels; to discard a security label, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Click the **SQL** tab to continue.

Your entries in the **Function** dialog generate a generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

### Example

The following is an example of the sql command generated by selections made in the **Function** dialog:



```

1 CREATE FUNCTION public.emp_count()
2     RETURNS bigint
3     LANGUAGE 'sql'
4     VOLATILE
5 AS $BODY$select count(emp_id) from employees;$BODY$;
6
7 ALTER FUNCTION public.emp_count()
8     OWNER TO postgres;
9
10 COMMENT ON FUNCTION public.emp_count()
11     IS 'This function gets the employee count';

```

The example demonstrates creating an **edbspl** function named **emp\_comp**. The function adds two columns (p\_sal and p\_comm), and then uses the result to compute a yearly salary, returning a NUMERIC value.

- Click the **Info** button (i) to access online help. View context-sensitive help in the **Tabbed browser**, where a new tab displays the PostgreSQL core documentation.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.

- Click the **Reset** button to restore configuration parameters.

### 34.5.11.15 Language Dialog

Use the CREATE LANGUAGE dialog to register a new procedural language.

The **Language** dialog organizes the registration of a procedural language through the following dialog tabs: **General**, **Definition**, and **Security**. The **SQL** tab displays the SQL code generated by dialog selections.

Create - Language

General

Definition


Security

SQL


Name


plperl


Owner


 postgres


Comment





 Cancel

 Reset

 Save

Use the fields in the **General** tab to identify a language:

- Use the drop-down listbox next to **Name** to select a language script.
- Use the drop-down listbox next to **Owner** to select a role.
- Store notes about the language in the **Comment** field.

Click the **Definition** tab to continue.

The screenshot shows a 'Create - Language' dialog box with four tabs: General, Definition, Security, and SQL. The 'Definition' tab is active. It contains a 'Trusted?' toggle switch set to 'Yes', and three empty drop-down menus for 'Handler function', 'Inline function', and 'Validator function'. At the bottom, there are buttons for 'Cancel', 'Reset', and 'Save' (which is highlighted with an orange border), along with information and help icons.

Use the fields in the **Definition** tab to define parameters:

- Move the **Trusted?** switch to the **No** position to specify only users with PostgreSQL superuser privilege can use this language. The default is **Yes**.
- When enabled, use the drop-down listbox next to **Handler Function** to select the function that will be called to execute the language's functions.
- When enabled, use the drop-down listbox next to **Inline Function** to select the function that will be called to execute an anonymous code block (DO command) in this language.
- When enabled, use the drop-down listbox next to **Validator Function** to select the function that will be called when a new function in the language is created, to validate the new function.

Click the **Security** tab to continue.

**Create - Language** [X]

General Definition **Security** SQL

**Privileges** +

| Grantee  | Privileges   | Grantor    |
|----------|--|------------|
| PUBLIC ▼ | <input checked="" type="checkbox"/> USAGE <input type="checkbox"/> WITH GRANT OPTION | postgres ▼ |

**Security labels** +

| Provider | Security label |
|----------|----------------|
|----------|----------------|

[i] [?] [X Cancel] [Reset] [Save]

Use the **Security** tab to assign privileges and define security labels.

Use the **Privileges** panel to assign privileges to a role. Click the **Add** icon (+) to set privileges for database objects:

- Select the name of the role from the drop-down listbox in the **Grantee** field.
- Click inside the **Privileges** field. Check the boxes to the left of one or more privileges to grant the selected privilege to the specified user.
- The current user, who is the default grantor for granting the privilege, is displayed in the **Grantor** field.

Click the **Add** icon (+) to assign additional privileges; to discard a privilege, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Use the **Security Labels** panel to define security labels applied to the function. Click the **Add** icon (+) to add each security label selection:

- Specify a security label provider in the **Provider** field. The named provider must be loaded and must consent to the proposed labeling operation.
- Specify a security label in the **Security Label** field. The meaning of a given label is at the discretion of the label provider. PostgreSQL places no restrictions on whether or how a label provider must interpret security labels; it merely provides a mechanism for storing them.

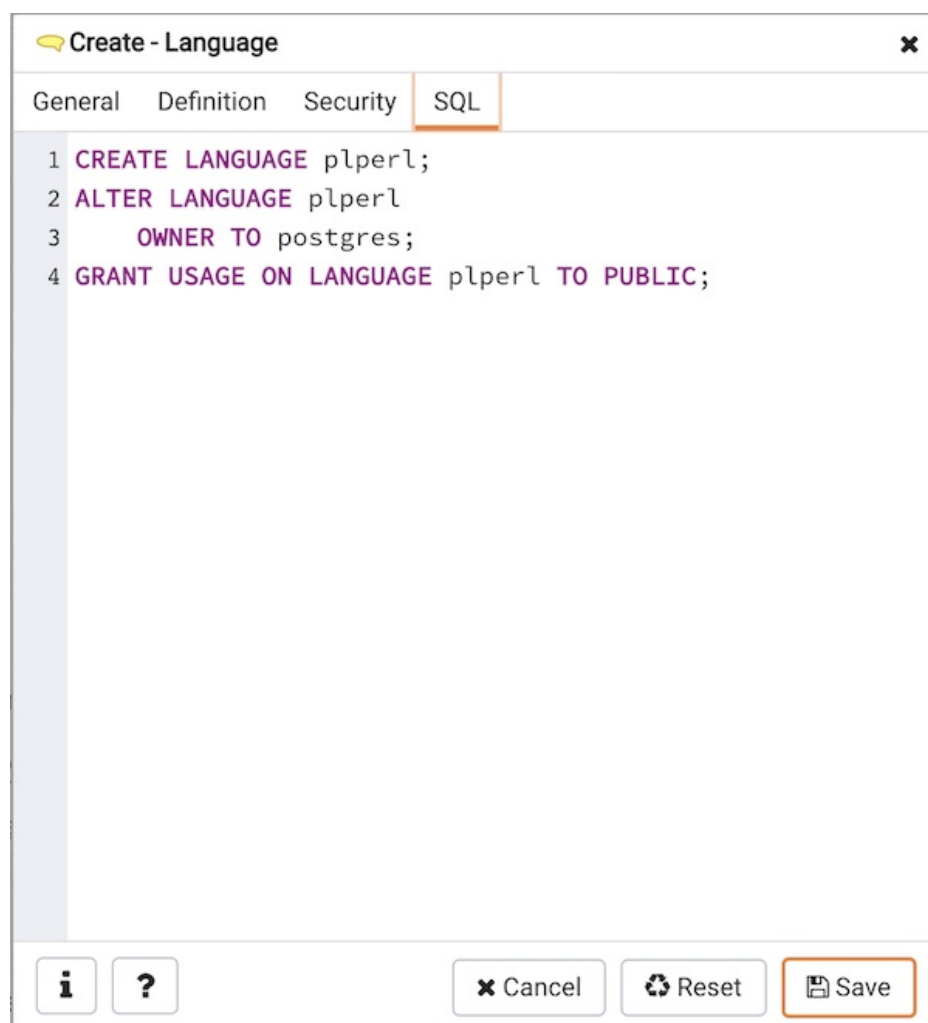
Click the **Add** icon (+) to assign additional security labels; to discard a security label, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Click the **SQL** tab to continue.

Your entries in the **Language** dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

#### Example

The following is an example of the sql command generated by user selections in the **Language** dialog:



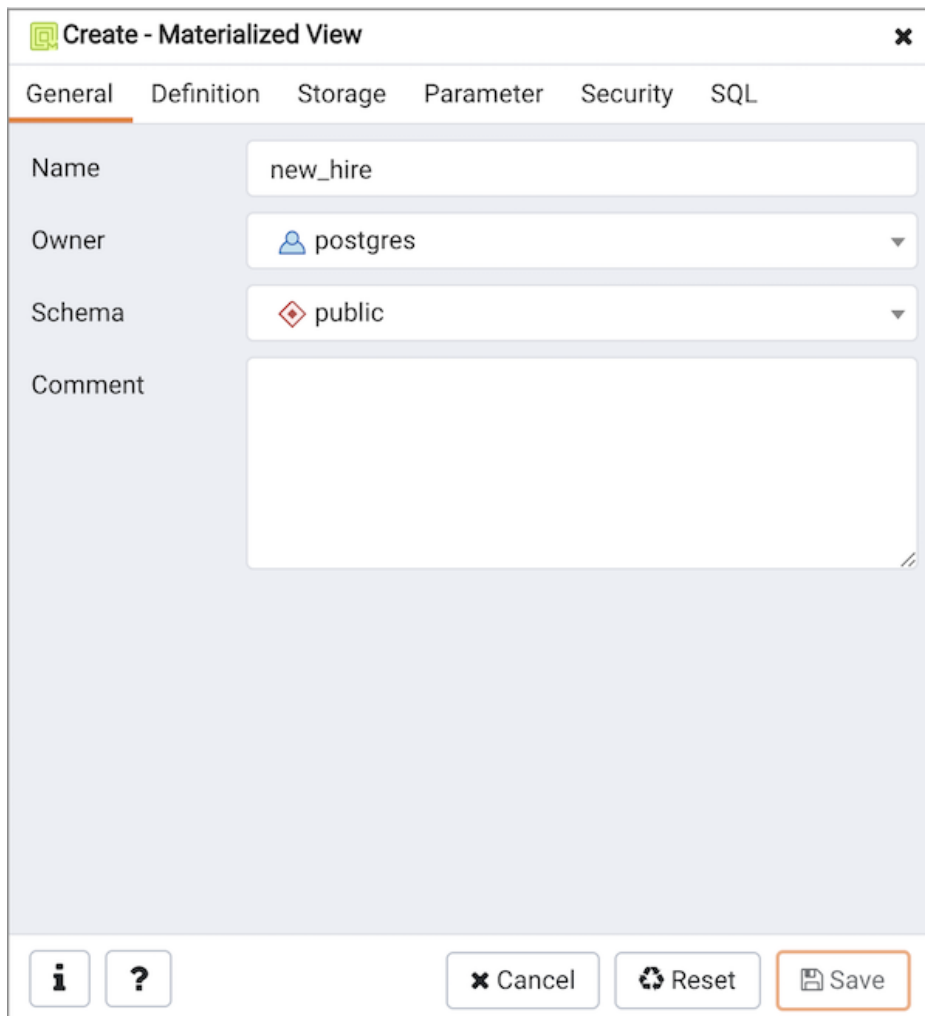
"The example shown demonstrates creating the procedural language named **plperl**."

- Click the **Info** button (i) to access online help. View context-sensitive help in the **Tabbed browser**, where a new tab displays the PostgreSQL core documentation.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

### 34.5.11.16 Materialized View Dialog

Use the **Materialized View** dialog to define a materialized view. A materialized view is a stored or cached view that contains the result set of a query. Use the REFRESH MATERIALIZED VIEW command to update the content of a materialized view.

The **Materialized View** dialog organizes the development of a materialized\_view through the following dialog tabs: **General**, **Definition**, **Storage**, **Parameter**, and **Security**. The **SQL** tab displays the SQL code generated by dialog selections.



The dialog box titled "Create - Materialized View" has a close button (X) in the top right corner. It features five tabs: "General", "Definition", "Storage", "Parameter", "Security", and "SQL". The "General" tab is selected and highlighted with an orange underline. Below the tabs, there are four fields: "Name" with a text input containing "new\_hire"; "Owner" with a dropdown menu showing a user icon and "postgres"; "Schema" with a dropdown menu showing a diamond icon and "public"; and "Comment" with a large, empty text area. At the bottom of the dialog, there are four buttons: an information icon (i), a question mark icon (?), a "Cancel" button with an X icon, a "Reset" button with a circular arrow icon, and a "Save" button with a floppy disk icon.

Use the fields in the **General** tab to identify the materialized view:

- Use the **Name** field to add a descriptive name for the materialized view. The name will be displayed in the **Browser** tree control.
- Use the drop-down listbox next to **Owner** to select the role that will own the materialized view.
- Select the name of the schema in which the materialized view will reside from the drop-down listbox in the **Schema** field.
- Store notes about the materialized view in the **Comment** field.

Click the **Definition** tab to continue.

Create - Materialized View

General

Definition

Storage

Parameter

Security

SQL

1

select \* from public.try;

i

?

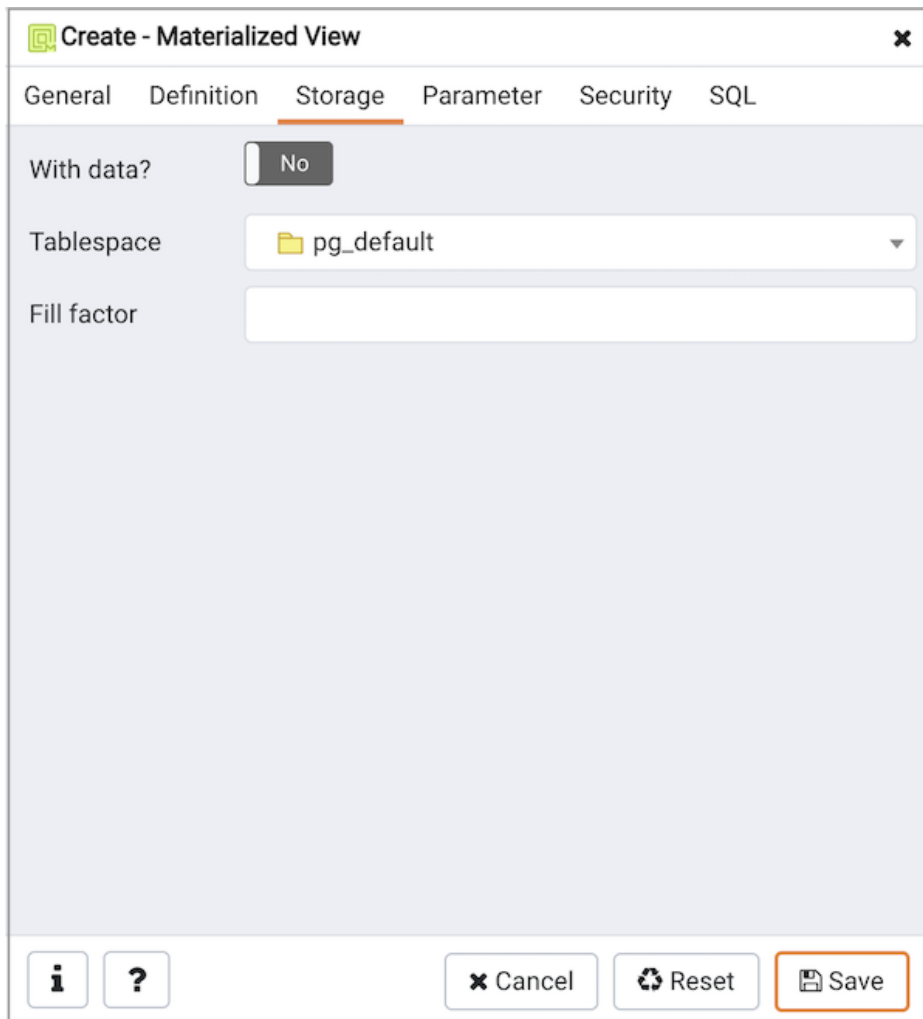
✕ Cancel

🔄 Reset

💾 Save

Use the text editor field in the **Definition** tab to provide the query that will populate the materialized view. Please note that updating the definition of existing materialized view would result in loss of Parameter(Table, Toast), Security(Privileges & Security labels), Indexes and other dependent objects.

Click the **Storage** tab to continue.



The image shows a 'Create - Materialized View' dialog box with a close button (X) in the top right corner. It has five tabs: 'General', 'Definition', 'Storage' (which is selected and underlined), 'Parameter', 'Security', and 'SQL'. The 'Storage' tab contains three fields: 'With data?' with a toggle switch currently set to 'No'; 'Tablespace' with a dropdown menu showing 'pg\_default'; and 'Fill factor' with an empty text input field. At the bottom of the dialog, there are four buttons: an information icon (i), a help icon (?), a 'Cancel' button with an X icon, a 'Reset' button with a circular arrow icon, and a 'Save' button with a floppy disk icon.

Use the fields in the **Storage** tab to maintain the materialized view:

- Move the **With Data** switch to the **Yes** position to specify the materialized view should be populated at creation time. If not, the materialized view cannot be queried until you invoke REFRESH MATERIALIZED VIEW.
- Use the drop-down listbox next to **Tablespace** to select a location for the materialized view.
- Use the **Fill Factor** field to specify a fill factor for the materialized view. The fill factor for a table is a percentage between 10 and 100. 100 (complete packing) is the default.

Click the **Parameter** tab to continue.



Create - Materialized View

General

Definition

Storage

Parameter

Security

SQL

Table

TOAST table

Custom auto-vacuum?

No

Autovacuum Enabled?

Not set

Yes

No

| Label                  | Value | Default     |
|------------------------|-------|-------------|
| ANALYZE scale factor   |       | 0.10        |
| ANALYZE base threshold |       | 50          |
| FREEZE maximum age     |       | 200,000,000 |
| VACUUM cost delay      |       | 20          |
| VACUUM cost limit      |       | -1          |
| VACUUM scale factor    |       | 0.20        |
| VACUUM base threshold  |       | 50          |
| FREEZE minimum age     |       | 50,000,000  |
| FREEZE table age       |       | 150,000,000 |

i

?

Cancel

Reset

Save

Use the tabs nested inside the **Parameter** tab to specify VACUUM and ANALYZE thresholds; use the **Table** tab and the **Toast Table** tab to customize values for the table and the associated toast table. To change the default values:

- Move the **Custom auto-vacuum?** switch to the **Yes** position to perform custom maintenance on the materialized view and to select values in the **Vacuum table** . The **Vacuum Table** provides default values for maintenance operations.
- Changing **Autovacuum enabled?** to **Not set** will reset autovacuum\_enabled.

Click the **Security** tab to continue.

Create - Materialized View

General

Definition

Storage

Parameter

Security

SQL

Privileges

Grantee

Privileges

Grantor

PUBLIC

☒ ALL

☐ WITH GRANT OPTION

☒ INSERT

☐ WITH GRANT OPTION

☒ SELECT

☐ WITH GRANT OPTION

☒ UPDATE

☐ WITH GRANT OPTION

☒ DELETE

☐ WITH GRANT OPTION

☒ TRUNCATE

☐ WITH GRANT OPTION

☒ REFERENCES

☐ WITH GRANT OPTION

☒ TRIGGER

☐ WITH GRANT OPTION

postgres

Security labels

Provider

Security label

i

?

Cancel

Reset

Save

Use the **Security** tab to assign privileges and define security labels.

Use the **Privileges** panel to assign privileges to a role. Click the **Add** icon (+) to set privileges for the materialized view:

- Select the name of the role from the drop-down listbox in the **Grantee** field.
- Click inside the **Privileges** field. Check the boxes to the left of one or more privileges to grant the selected privilege to the specified user.
- The current user, who is the default grantor for granting the privilege, is displayed in the **Grantor** field.

Click the **Add** icon (+) to assign additional privileges; to discard a privilege, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Use the **Security Labels** panel to define security labels applied to the materialized view. Click the **Add** icon (+) to add each security label selection:

- Specify a security label provider in the **Provider** field. The named provider must be loaded and must consent to the proposed labeling operation.
- Specify a security label in the **Security Label** field. The meaning of a given label is at the discretion of the label provider. PostgreSQL places no restrictions on whether or how a label provider must interpret security labels; it merely provides a mechanism for storing them.

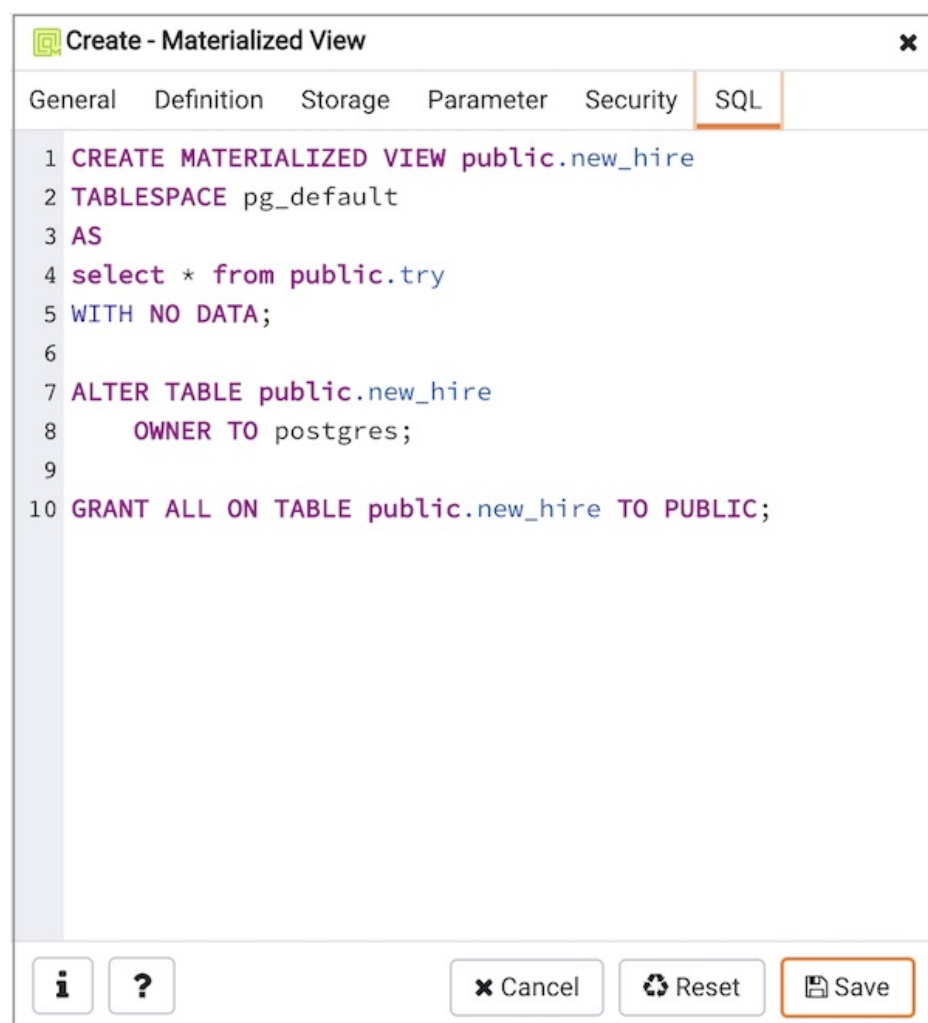
Click the **Add** icon (+) to assign additional security labels; to discard a security label, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Click the **SQL** tab to continue.

Your entries in the **Materialized View** dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

Example

The following is an example of the sql command generated by user selections in the **Materialized View** dialog:



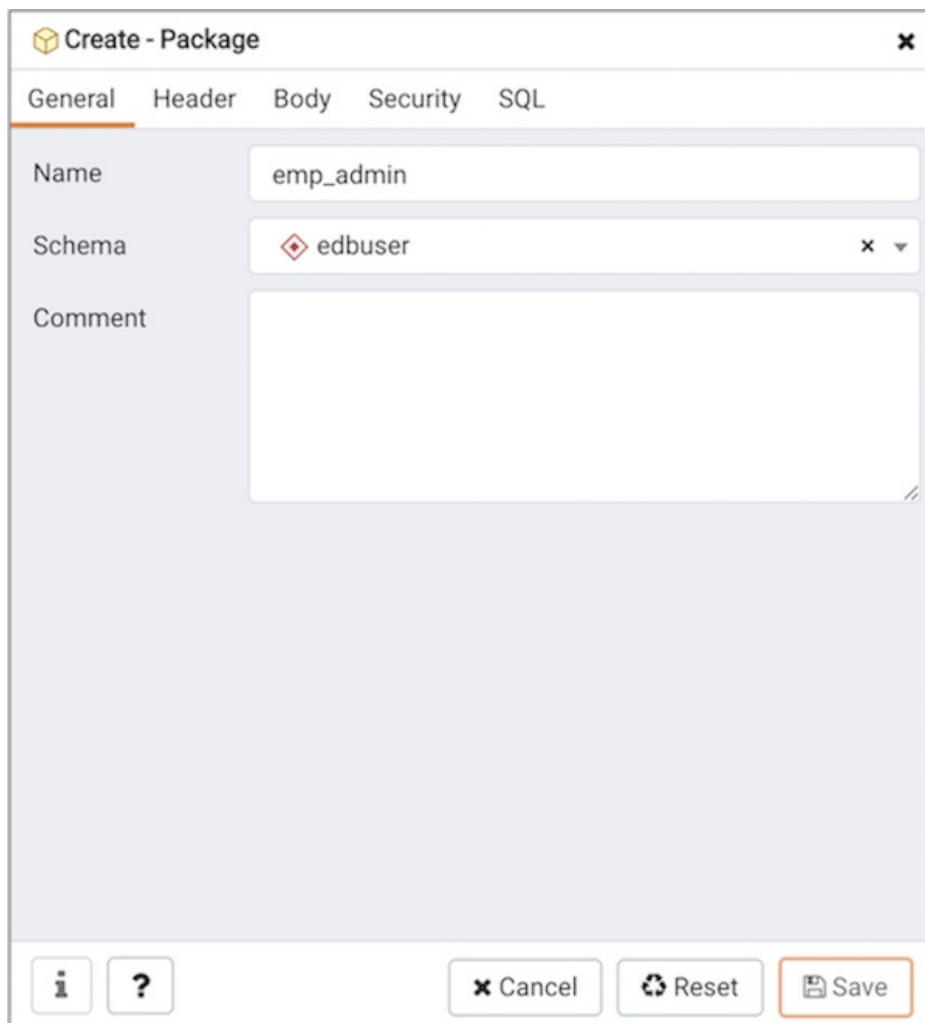
The example shown creates a query named **new\_hires** that stores the result of the displayed query in the **pg\_default** tablespace.

- Click the **Info** button (i) to access online help. View context-sensitive help in the **Tabbed browser**, where a new tab displays the PostgreSQL core documentation.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

### 34.5.11.17 Package Dialog

Use the **Package** dialog to create a (user-defined) package specification.

The **Package** dialog organizes the management of a package through the following dialog tabs: **General**, **Header**, **Body**, and **Security**. The **SQL** tab displays the SQL code generated by dialog selections.

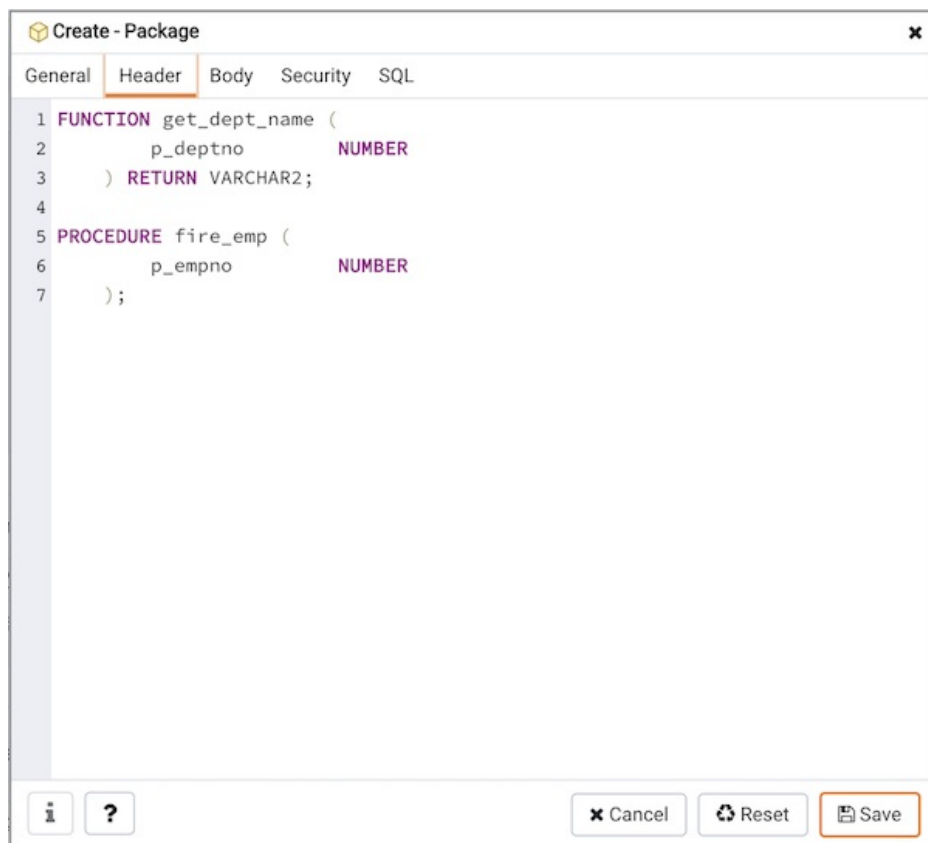


The image shows a 'Create - Package' dialog box with a title bar containing a cube icon and a close button. Below the title bar are five tabs: 'General', 'Header', 'Body', 'Security', and 'SQL'. The 'General' tab is selected and highlighted with an orange underline. The 'General' tab contains three fields: 'Name' with the text 'emp\_admin', 'Schema' with a dropdown menu showing 'edbuser' and a close button, and a large 'Comment' text area. At the bottom of the dialog are four buttons: an information icon, a question mark icon, a 'Cancel' button with a close icon, a 'Reset' button with a circular arrow icon, and a 'Save' button with a disk icon and an orange border.

Use the fields in the **General** tab to identify the package:

- Use the **Name** field to add a descriptive name for the package. The name of a new package must not match any existing package in the same schema.
- Select the schema in which the package will reside from the drop-down listbox in the **Schema** field.
- Store notes about the package in the **Comment** field.

Click the **Header** tab to continue.



The 'Create - Package' dialog box is shown with the 'Header' tab selected. The 'General' tab is also visible. The 'Header' tab contains the following SQL code:

```

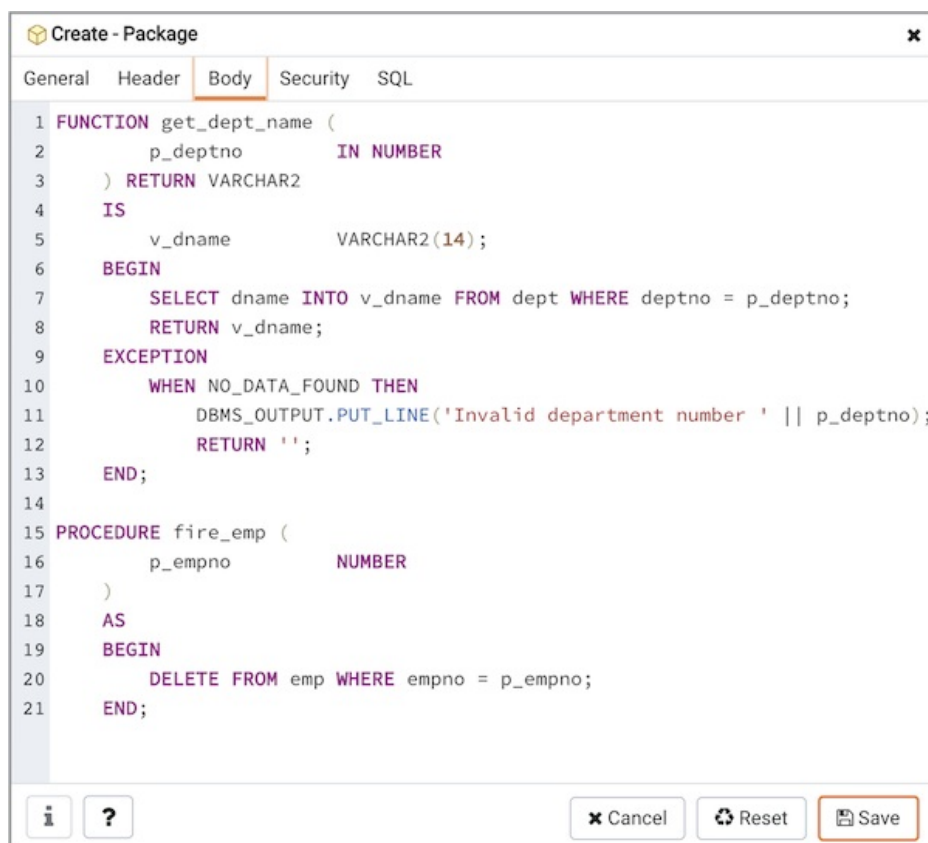
1 FUNCTION get_dept_name (
2     p_deptno      NUMBER
3 ) RETURN VARCHAR2;
4
5 PROCEDURE fire_emp (
6     p_empno       NUMBER
7 );

```

At the bottom of the dialog, there are buttons for 'Cancel', 'Reset', and 'Save'.

Use the **Header** field to define the public interface for the package.

Click the **Body** tab to continue.



The 'Create - Package' dialog box is shown with the 'Body' tab selected. The 'General' and 'Header' tabs are also visible. The 'Body' tab contains the following SQL code:

```

1 FUNCTION get_dept_name (
2     p_deptno      IN NUMBER
3 ) RETURN VARCHAR2
4 IS
5     v_dname       VARCHAR2(14);
6 BEGIN
7     SELECT dname INTO v_dname FROM dept WHERE deptno = p_deptno;
8     RETURN v_dname;
9 EXCEPTION
10    WHEN NO_DATA_FOUND THEN
11        DBMS_OUTPUT.PUT_LINE('Invalid department number ' || p_deptno);
12        RETURN '';
13 END;
14
15 PROCEDURE fire_emp (
16     p_empno       NUMBER
17 )
18 AS
19 BEGIN
20     DELETE FROM emp WHERE empno = p_empno;
21 END;

```

At the bottom of the dialog, there are buttons for 'Cancel', 'Reset', and 'Save'.

Use the **Body** field to provide the code that implements each package object.

Click the **Security** tab to continue.

Create - Package

GeneralHeaderBodySecuritySQL

Privileges

| Grantee | Privileges | Grantor |
|---------|------------|---------|
|         |            |         |

Cancel

Reset

Save

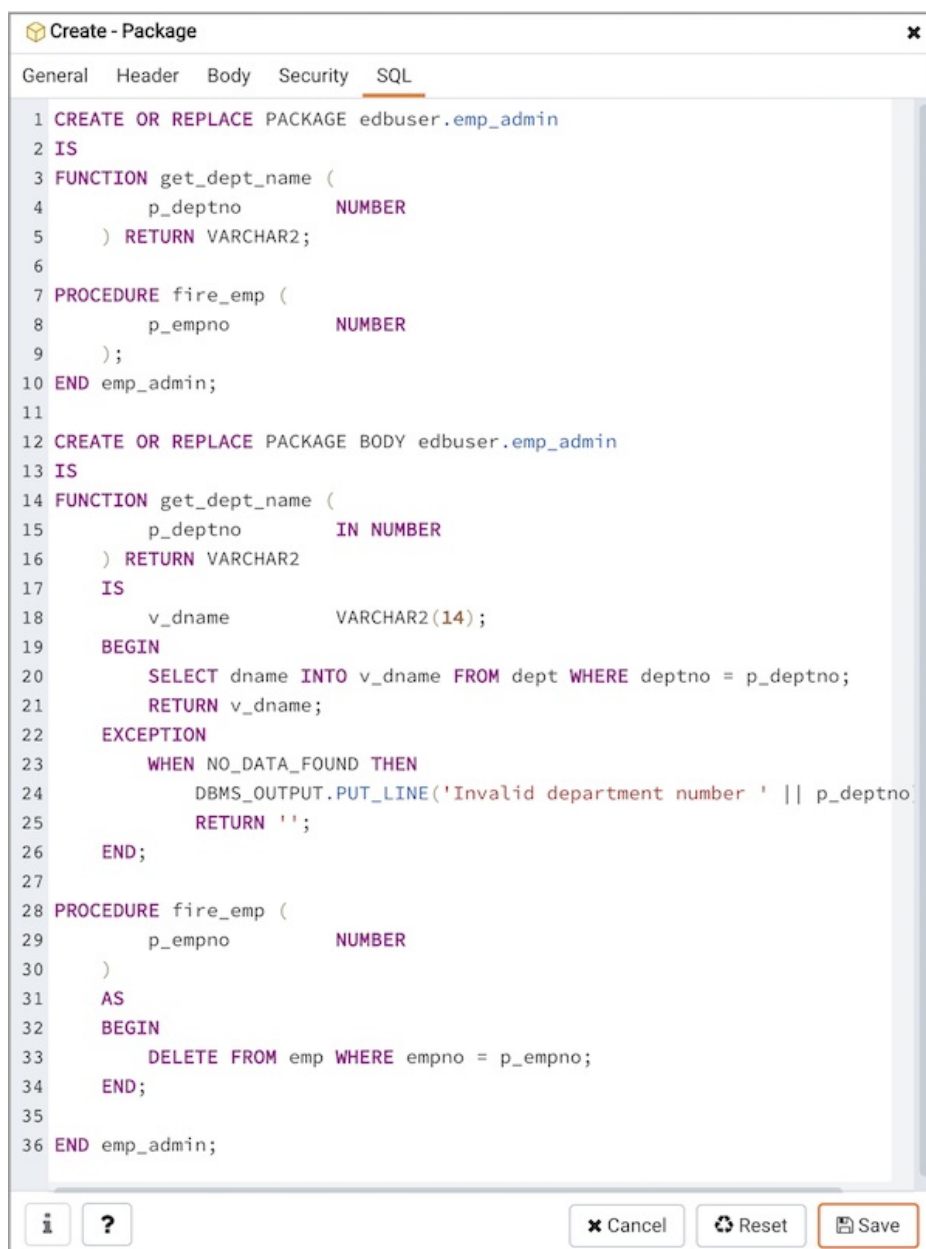
Use the fields in the **Security** tab to assign EXECUTE privileges for the package to a role. Click the **Add** icon (+) to set privileges for the package:

- Select the name of the role from the drop-down listbox in the **Grantee** field.
- Click inside the **Privileges** field. Check the boxes to the left of a privilege to grant the selected privilege to the specified user.
- The current user, who is the default grantor for granting the privilege, is displayed in the **Grantor** field.

Click the **Add** icon (+) to assign additional privileges; to discard a privilege, click the trash icon to the left of the row, and confirm the deletion in the **Delete Row** popup.

Click the **SQL** tab to continue.

Your entries in the **Package** dialog generate a SQL command that creates or modifies a package definition:



The example shown demonstrates creating a package named `empinfo` that includes two function and two procedure.

- Click the `Save` button to save work.
- Click the `Cancel` button to exit without saving work.
- Click the `Reset` button to delete any changes to the dialog.

### 34.5.11.18 Procedure Dialog

Use the `Procedure` dialog to create a procedure; procedures are supported by PostgreSQL v11+ and EDB Postgres Advanced Server. The `Procedure` dialog allows you to implement options of the CREATE PROCEDURE command.

The `Procedure` dialog organizes the development of a procedure through the following dialog tabs: `General`, `Definition`, `Options`, `Arguments`, `Parameters`, and `Security`. The `SQL` tab displays the SQL code generated by dialog selections.

Create - Procedure

GeneralDefinitionCodeOptionsParametersSecuritySQL

Name

insert\_data

Owner

postgres

Schema

public

Comment

i?

CancelResetSave

Use the fields in the **General** tab to identify a procedure:

- Use the **Name** field to add a descriptive name for the procedure. The name will be displayed in the \*\* tree control.
- Use the drop-down listbox next to **Owner** to select a role.
- Select the name of the schema in which the procedure will reside from the drop-down listbox in the **Schema** field.
- Store notes about the procedure in the **Comment** field.

Click the **Definition** tab to continue.

Create - Procedure

GeneralDefinitionCodeOptionsParametersSecuritySQL

Language

sql

Arguments

i?

CancelResetSave

Use the fields in the **Definition** tab to define the procedure:

- Use the drop-down listbox next to **Language** to select a language. The default is **edbspl**.



- Use the fields in the **Arguments** section to define an argument. Click **Add** to set parameters and values for the argument:
- Use the drop-down listbox next to **Data type** to select a data type.
- Use the drop-down listbox next to **Mode** to select a mode. Select **IN** for an input parameter; select **OUT** for an output parameter; select **INOUT** for both an input and an output parameter; or, select **VARIADIC** to specify a VARIADIC parameter.
- Write a name for the argument in the **Argument Name** field.
- Specify a default value for the argument in the **Default Value** field.

Click **Add** to define another argument; to discard an argument, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Click the **Code** tab to continue.

**Create - Procedure**

General Definition **Code** Options Parameters Security SQL

```

1 INSERT INTO public.try VALUES (70);
2 INSERT INTO public.try VALUES (80);

```

Buttons: **Cancel** **Reset** **Save**

- Use the **Code** field to specify the code that will execute when the procedure is called.

Click the **Options** tab to continue.

Use the fields in the **Options** tab to describe or modify the behavior of the procedure:

- Use the drop-down listbox under **Volatility** to select one of the following. **VOLATILE** is the default value.
  - **VOLATILE** indicates that the value can change even within a single table scan, so no optimizations can be made.
  - **STABLE** indicates that the procedure cannot modify the database, and that within a single table scan it will consistently return the same result for the same argument values, but that its result could change across SQL statements.
  - **IMMUTABLE** indicates that the procedure cannot modify the database and always returns the same result when given the same argument values.
- Move the **Strict?** switch to indicate if the procedure always returns NULL whenever any of its arguments are NULL. If **Yes**, the procedure is not executed when there are NULL arguments; instead a NULL result is assumed automatically. The default is **No**.
- Move the **Security of definer?** switch to specify that the procedure is to be executed with the privileges of the user that created it. The default is **No**.
- Use the **Estimated cost** field to specify a positive number representing the estimated execution cost for the procedure, in units of `cpu_operator_cost`. If the procedure returns a set, this is the cost per returned row.
- Move the **Leak proof?** switch to indicate whether the procedure has side effects — it reveals no information about its arguments other than by its return value. The default is **No**.

Click the **Parameters** tab to continue.

Create - Procedure

General

Definition

Code

Options

Parameters

Security

SQL

| Name | Value |
|------|-------|
|------|-------|

Cancel

Reset

Save

Use the fields in the **Parameters** tab to specify settings that will be applied when the procedure is invoked:

- Use the drop-down listbox next to **Parameter Name** in the **Parameters** panel to select a parameter.
- Click the **Add** button to add the variable to **Name** field in the table.
- Use the **Value** field to specify the value that will be associated with the selected variable. This field is context-sensitive.

Click the **Security** tab to continue.

Create - Procedure

General

Definition

Code

Options

Parameters

Security

SQL

Privileges

| Grantee                        | Privileges   | Grantor                        |
|--------------------------------|--|--------------------------------|
| <div><div></div>postgres</div> | <div><div><input checked="" type="checkbox"/> EXECUTE</div><div><input type="checkbox"/> WITH GRANT OPTION</div></div> | <div><div></div>postgres</div> |

Cancel

Reset

Save

Use the **Security** tab to assign privileges and define security labels.

Use the **Privileges** panel to assign execute privileges for the procedure to a role:

- Select the name of the role from the drop-down listbox in the **Grantee** field.
- Click inside the **Privileges** field. Check the boxes to the left of one or more privileges to grant the selected privilege to the specified user.
- The current user, who is the default grantor for granting the privilege, is displayed in the **Grantor** field.

Click **Add** to assign additional privileges; to discard a privilege, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Use the **Security Labels** panel to define security labels applied to the procedure. Click **Add** to add each security label selection:

- Specify a security label provider in the **Provider** field. The named provider must be loaded and must consent to the proposed labeling operation.
- Specify a security label in the **Security Label** field. The meaning of a given label is at the discretion of the label provider. PostgreSQL places no restrictions on whether or how a label provider must interpret security labels; it merely provides a mechanism for storing them.

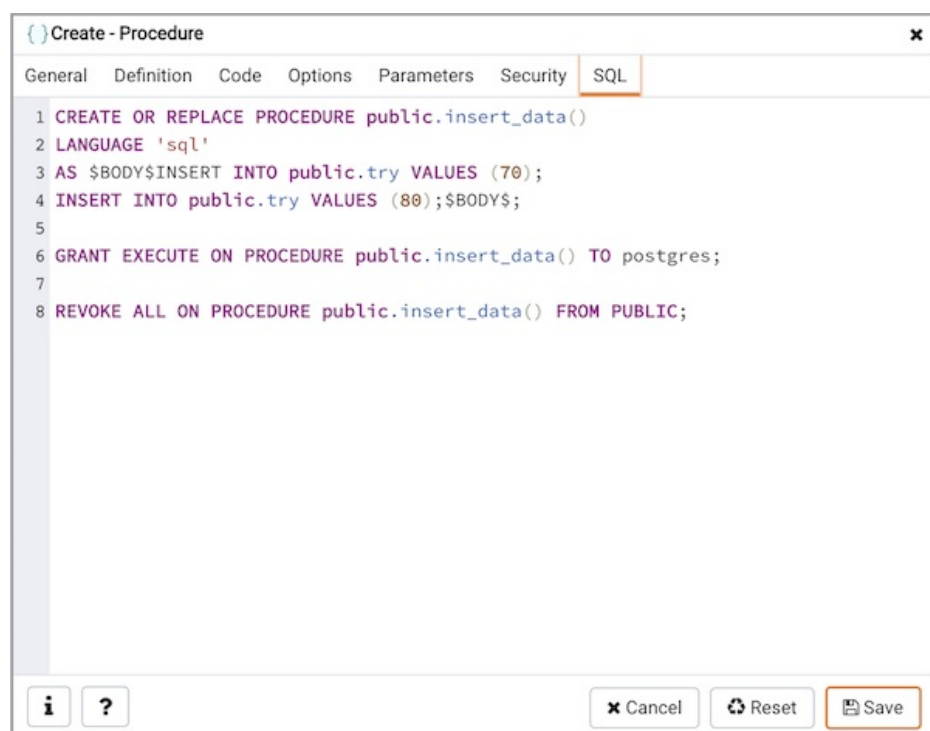
Click **Add** to assign additional security labels; to discard a security label, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Click the **SQL** tab to continue.

Your entries in the **Procedure** dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

### Example

The following is an example of the sql command generated by selections made in the **Procedure** dialog:



```

1 CREATE OR REPLACE PROCEDURE public.insert_data()
2 LANGUAGE 'sql'
3 AS $BODY$INSERT INTO public.try VALUES (70);
4 INSERT INTO public.try VALUES (80);$BODY$;
5
6 GRANT EXECUTE ON PROCEDURE public.insert_data() TO postgres;
7
8 REVOKE ALL ON PROCEDURE public.insert_data() FROM PUBLIC;

```

The example demonstrates creating a procedure that returns a list of employees from a table named **emp**. The procedure is a SECURITY DEFINER, and will execute with the privileges of the role that defined the procedure.

- Click the **Info** button (i) to access online help. View context-sensitive help in the **Tabbed browser**, where a new tab displays the PostgreSQL core documentation.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

### 34.5.11.19 Publication Dialog

Logical replication uses a **publish** and **subscribe** model with one or more **subscribers** subscribing to one or more **publications** on a publisher node.

Use the **publication** dialog to create a publication. A publication is a set of changes generated from a table or a group of tables, and might also be described as a change set or replication set.

The **publication** dialog organizes the development of a publication through the following dialog tabs: **General** and **Definition**. The **SQL** tab displays the SQL code generated by dialog selections.

The screenshot shows a 'Create - Publication' dialog box. It has three tabs: 'General', 'Definition', and 'SQL'. The 'General' tab is active. It contains two input fields: 'Name' with the text 'pub1' and 'Owner' with a dropdown menu showing 'postgres'. At the bottom, there are buttons for 'Cancel', 'Reset', and 'Save' (highlighted with an orange border), along with information and help icons.

Use the fields in the **General** tab to identify the publication:

- Use the **Name** field to add a descriptive name for the publication. The name will be displayed in the Browser tree control.
- The **Owner** field takes the name of the user automatically as current connected user. You can change the owner after creating publication using alter publication.

Click the **Definition** tab to continue.

**Create - Publication**

General **Definition** SQL

All tables? ☐ No

Only table? ☐ No

If ONLY is specified before the table name, only that table is added to the publication. If ONLY is not specified, the table and all its descendant tables (if any) are added.

Tables:

With

|        |   |          |   |
|--------|---|----------|---|
| INSERT | <input checked="" type="checkbox"/> Yes | UPDATE   | <input checked="" type="checkbox"/> Yes |
| DELETE | <input checked="" type="checkbox"/> Yes | TRUNCATE | <input checked="" type="checkbox"/> Yes |

**Buttons:** [i] [?] [Cancel] [Reset] [Save]

Use the **Definition** tab to set properties for the publication:

- Move the switch next to **All tables?** to **Yes** to replicate all the tables of the database, including tables created in the future.
- Move the switch next to **Only table?** to **Yes** to replicate only the listed tables excluding all its descendant tables.
- Specify a table or list of tables separated by a comma in **Tables** field to replicate all the listed table.
- Use the **With** section to determine which DML operations will be published by the new publication to the subscribers. Move the switch next to **INSERT**, **UPDATE**, **DELETE**, or **TRUNCATE** to **No** if you do not want to replicate any of these DML operations from Publication to Subscription. By default, all the switches are set to **Yes** allowing all the DML operations.

#### Note

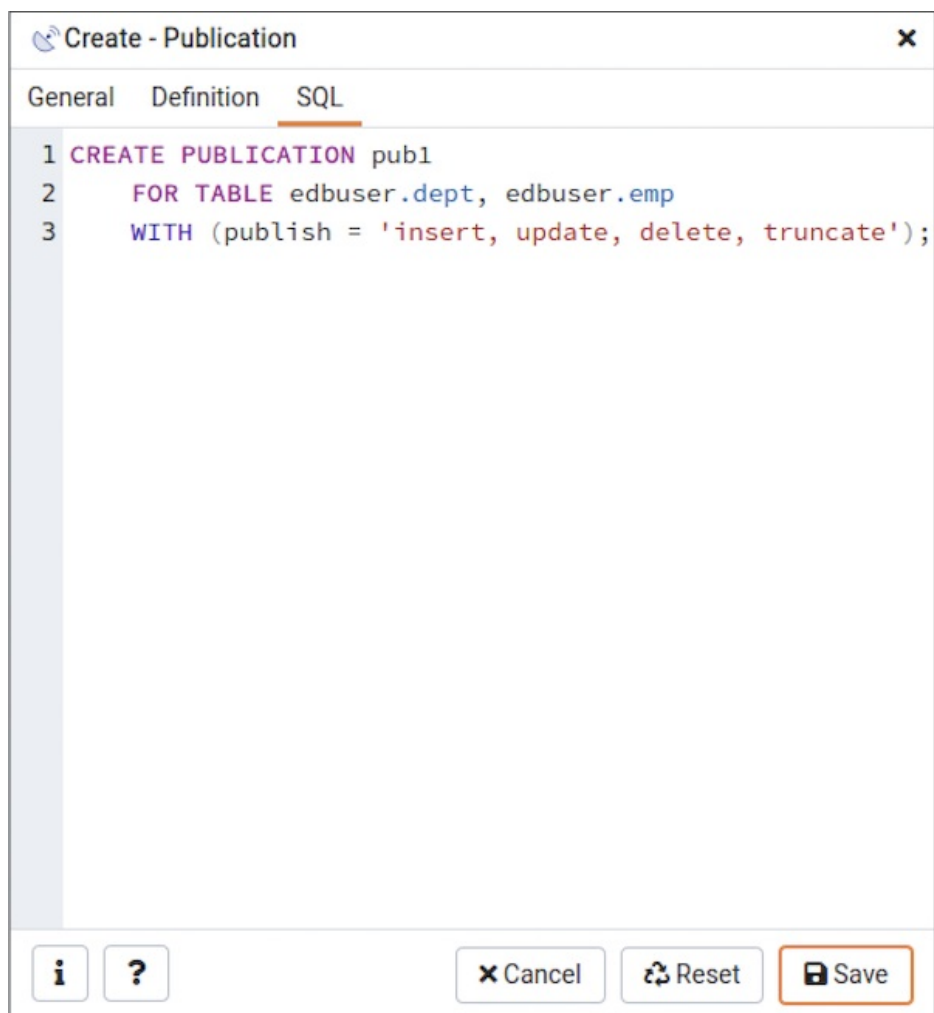
A published table must have a **replica identity** configured in order to be able to replicate UPDATE and DELETE operations. You can change with ALTER TABLE statement. For more information on replica identity see [Logical Replication Publication](#).

Click the **SQL** tab to continue.

Your entries in the **Publication** dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

#### Example

The following is an example of the sql command generated by user selections in the **Publication** dialog:



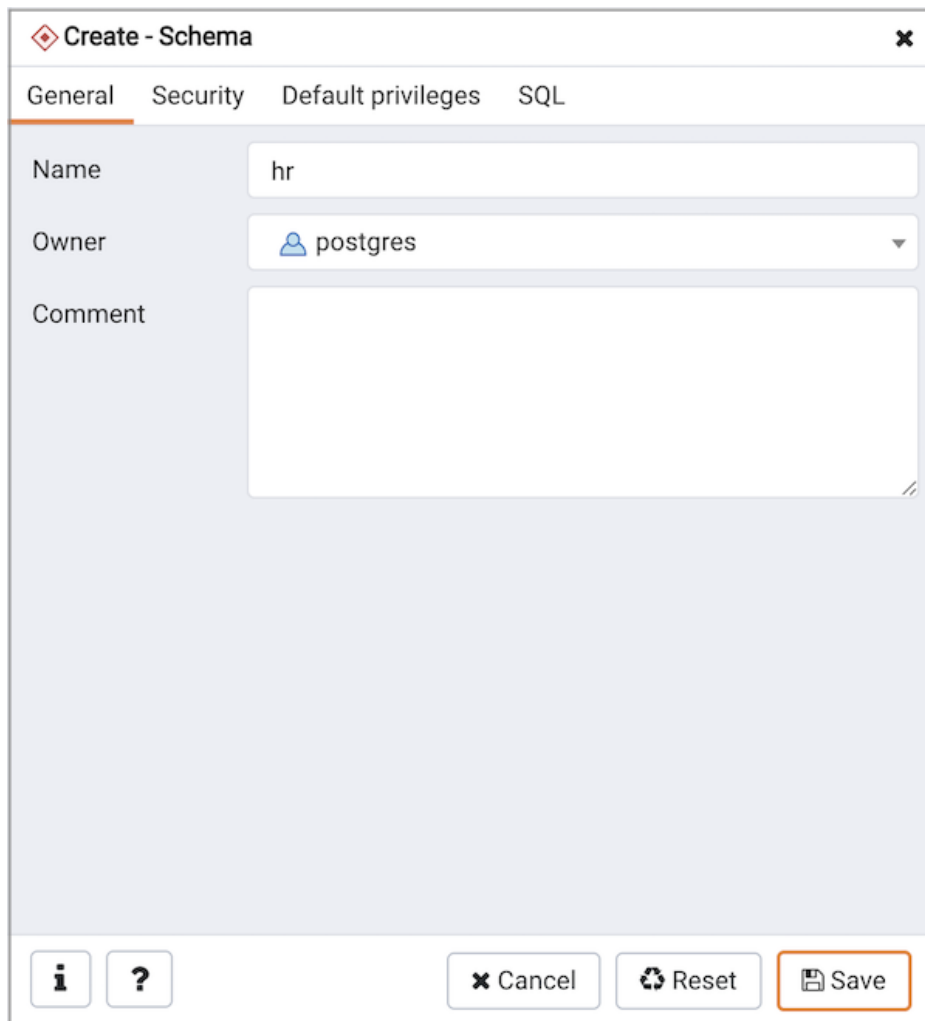
The example creates a publication named `pub1` that is owned by `postgres`. It allows replication of all the DML operations.

- Click the **Info** button (i) to access online help.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore all the default settings.

### 34.5.11.20 Schema Dialog

Use the **Schema** dialog to define a schema. A schema is the organizational workhorse of a database, similar to directories or namespaces. To create a schema, you must be a database superuser or have the CREATE privilege.

The **Schema** dialog organizes the development of schema through the following dialog tabs: **General** and **Security**. The **SQL** tab displays the SQL code generated by dialog selections.

A screenshot of the 'Create - Schema' dialog box in Postgres Enterprise Manager. The dialog has a title bar with a red diamond icon and a close button. Below the title bar are four tabs: 'General' (selected), 'Security', 'Default privileges', and 'SQL'. The 'General' tab contains three fields: 'Name' with the value 'hr', 'Owner' with a dropdown menu showing 'postgres' and a user icon, and 'Comment' with a large empty text area. At the bottom of the dialog are four buttons: an information icon, a question mark icon, a 'Cancel' button, a 'Reset' button, and a 'Save' button which is highlighted with an orange border.

Use the fields on the **General** tab to identify the schema.

- Use the **Name** field to add a descriptive name for the schema. The name will be displayed in the **Browser** tree control.
- Select the owner of the schema from the drop-down listbox in the **Owner** field.
- Store notes about the schema in the **Comment** field.

Click the **Security** tab to continue.



**Create - Schema**

General **Security** Default privileges SQL

**Privileges** +

| Grantee   | Privileges                                 | Grantor  |
|-----------|--|----------|
| pem_admin | <input type="checkbox"/> ALL               | postgres |
|           | <input type="checkbox"/> WITH GRANT OPTION |          |
|           | <input checked="" type="checkbox"/> CREATE |          |
|           | <input checked="" type="checkbox"/> USAGE  |          |

**Security labels** +

| Provider | Security label |
|----------|----------------|
|----------|----------------|

Cancel
 Reset
 Save

Use the **Security** tab to assign privileges and security labels for the schema.

Click the **Add** icon (+) to assign a set of privileges in the **Privileges** panel:

- Select the name of the role from the drop-down listbox in the **Grantee** field.
- Click inside the **Privileges** field. Check the boxes to the left of one or more privileges to grant the selected privileges to the specified user.
- Select the name of the role that is granting the privilege from the drop-down listbox in the **Grantor** field. The default grantor is the owner of the schema.

Click the **Add** icon (+) to assign additional sets of privileges; to discard a privilege, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Click the **Add** icon (+) to assign a security label in the **Security Labels** panel:

- Specify a security label provider in the **Provider** field. The named provider must be loaded and must consent to the proposed labeling operation.
- Specify a security label in the **Security Label** field. The meaning of a given label is at the discretion of the label provider. PostgreSQL places no restrictions on whether or how a label provider must interpret security labels; it merely provides a mechanism for storing them.

Click the **Add** icon (+) to assign additional security labels; to discard a security label, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Click the **Default Privileges** tab to continue.

Create - Schema

General

Security

Default privileges

SQL

Tables

Sequences

Functions

Types

| Grantee   | Privileges  | Grantor   |
|---|---|---|
| <div><div></div><div><div><div></div><div>pem_admin</div></div></div></div> | <div><div><input checked="" type="checkbox"/></div>ALL</div>        | <div><div><input type="checkbox"/></div>WITH GRANT OPTION</div> |
|   | <div><div><input checked="" type="checkbox"/></div>INSERT</div>     | <div><div><input type="checkbox"/></div>WITH GRANT OPTION</div> |
|   | <div><div><input checked="" type="checkbox"/></div>SELECT</div>     | <div><div><input type="checkbox"/></div>WITH GRANT OPTION</div> |
|   | <div><div><input checked="" type="checkbox"/></div>UPDATE</div>     | <div><div><input type="checkbox"/></div>WITH GRANT OPTION</div> |
|   | <div><div><input checked="" type="checkbox"/></div>DELETE</div>     | <div><div><input type="checkbox"/></div>WITH GRANT OPTION</div> |
|   | <div><div><input checked="" type="checkbox"/></div>TRUNCATE</div>   | <div><div><input type="checkbox"/></div>WITH GRANT OPTION</div> |
|   | <div><div><input checked="" type="checkbox"/></div>REFERENCES</div> | <div><div><input type="checkbox"/></div>WITH GRANT OPTION</div> |
|   | <div><div><input checked="" type="checkbox"/></div>TRIGGER</div>    | <div><div><input type="checkbox"/></div>WITH GRANT OPTION</div> |

?

Cancel

Reset

Save

Use the **Default Privileges** tab to grant privileges for tables, sequences, functions and types. Use the tabs nested inside the **Default Privileges** tab to specify the database object and click the **Add** icon (+) to assign a set of privileges:

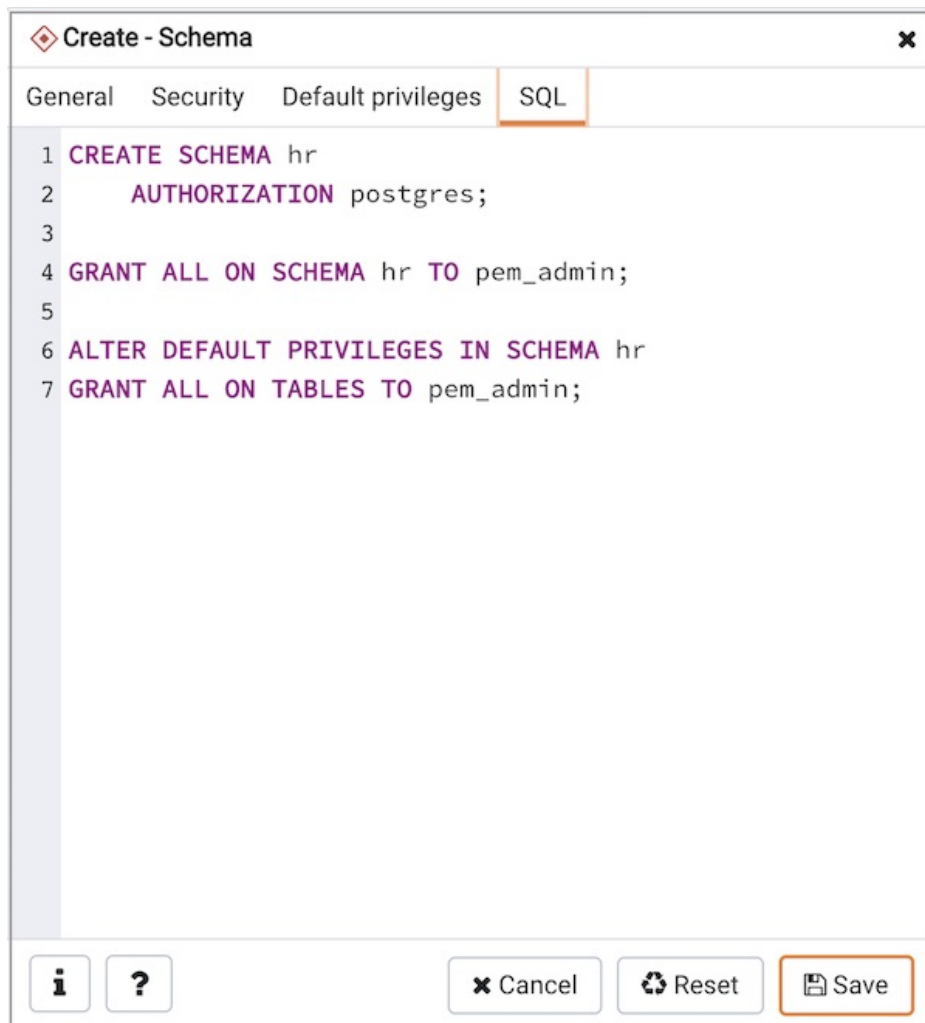
- Select the name of a role that will be granted privileges in the schema from the drop-down listbox in the **Grantee** field.
- Click inside the **Privileges** field. Check the boxes to the left of one or more privileges to grant the selected privileges to the specified user.
- The current user, who is the default grantor for granting the privilege, is displayed in the **Grantor** field.

Click the **SQL** tab to continue.

Your entries in the **Schema** dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

### Example

The following is an example of the sql command generated by selections made in the **Schema** dialog:



The example creates a schema named `hr`; the command grants `USAGE` privileges to `public` and assigns the ability to grant privileges to `alice`.

- Click the **Info** button (i) to access online help. View context-sensitive help in the **Tabbed browser**, where a new tab displays the PostgreSQL core documentation.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

### 34.5.11.21 Sequence Dialog

Use the **Sequence** dialog to create a sequence. A sequence generates unique values in a sequential order (not necessarily contiguous).

The **Sequence** dialog organizes the development of a sequence through the following dialog tabs: **General**, **Definition**, and **Security**. The **SQL** tab displays the SQL code generated by dialog selections.


**1.3 Create - Sequence** ✕

General Definition Security SQL


Name

seq1



Owner



 postgres ✕ ▼

Schema

 public ✕ ▼

Comment

✕ Cancel  Reset  Save

Use the fields in the **General** tab to identify a sequence:

- Use the **Name** field to add a descriptive name for the sequence. The name will be displayed in the **Browser** tree control. The sequence name must be distinct from the name of any other sequence, table, index, view, or foreign table in the same schema.
- Use the drop-down listbox next to **Owner** to select the name of the role that will own the sequence.
- Use the drop-down listbox next to **Schema** to select the schema in which the sequence will reside.
- Store notes about the sequence in the **Comment** field.

Click the **Definition** tab to continue.

1.3 Create - Sequence

General
Definition
Security
SQL

Increment
10

Start
1

Minimum

Maximum
100

Cache

Cycled
No

i
?
x Cancel
Reset
Save

Use the fields in the **Definition** tab to define the sequence:

- Use the **Increment** field to specify which value is added to the current sequence value to create a new value.
- Provide a value in the **Start** field to specify the beginning value of the sequence. The default starting value is MINVALUE for ascending sequences and MAXVALUE for descending ones.
- Provide a value in the **Minimum** field to specify the minimum value a sequence can generate. If this clause is not supplied or NO MINVALUE is specified, then defaults will be used. The defaults are 1 and -263-1 for ascending and descending sequences, respectively.
- Provide a value in the **Maximum** field to specify the maximum value for the sequence. If this clause is not supplied or NO MAXVALUE is specified, then default values will be used. The defaults are 263-1 and -1 for ascending and descending sequences, respectively.
- Provide a value in the **Cache** field to specify how many sequence numbers are to be preallocated and stored in memory for faster access. The minimum value is 1 (only one value can be generated at a time, i.e., no cache), and this is also the default.
- Move the **Cycled** switch to the **Yes** position to allow the sequence to wrap around when the MAXVALUE or the MINVALUE has been reached by an ascending or descending sequence respectively. If the limit is reached, the next number generated will be the MINVALUE or MAXVALUE, respectively. The default is **No**.

Click the **Security** tab to continue.

1.3 Create - Sequence

General

Definition

Security

SQL

Privileges

Grantee

Privileges

Grantor

PUBLIC

ALL

WITH GRANT OPTION

SELECT

WITH GRANT OPTION

UPDATE

WITH GRANT OPTION

USAGE

WITH GRANT OPTION

postgres

Security labels

Provider

Security label

i

?

Cancel

Reset

Save

Use the **Security** tab to assign privileges and define security labels for the sequence.

Use the **Privileges** panel to assign privileges. Click the **Add** icon (+) to set privileges:

- Select the name of a role that will be granted privileges from the drop-down listbox in the **Grantee** field.
- Click inside the **Privileges** field. Check the boxes to the left of one or more privileges to grant the selected privilege to the specified user.
- The current user, who is the default grantor for granting the privilege, is displayed in the **Grantor** field.

Click the **Add** icon (+) to assign additional privileges; to discard a privilege, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Use the **Security Labels** panel to define security labels applied to the sequence. Click the **Add** icon (+) to add each security label selection:

- Specify a security label provider in the **Provider** field. The named provider must be loaded and must consent to the proposed labeling operation.
- Specify a security label in the **Security Label** field. The meaning of a given label is at the discretion of the label provider. PostgreSQL places no restrictions on whether or how a label provider must interpret security labels; it merely provides a mechanism for storing them.

Click the **Add** icon (+) to assign additional security labels; to discard a security label, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Click the **SQL** tab to continue.

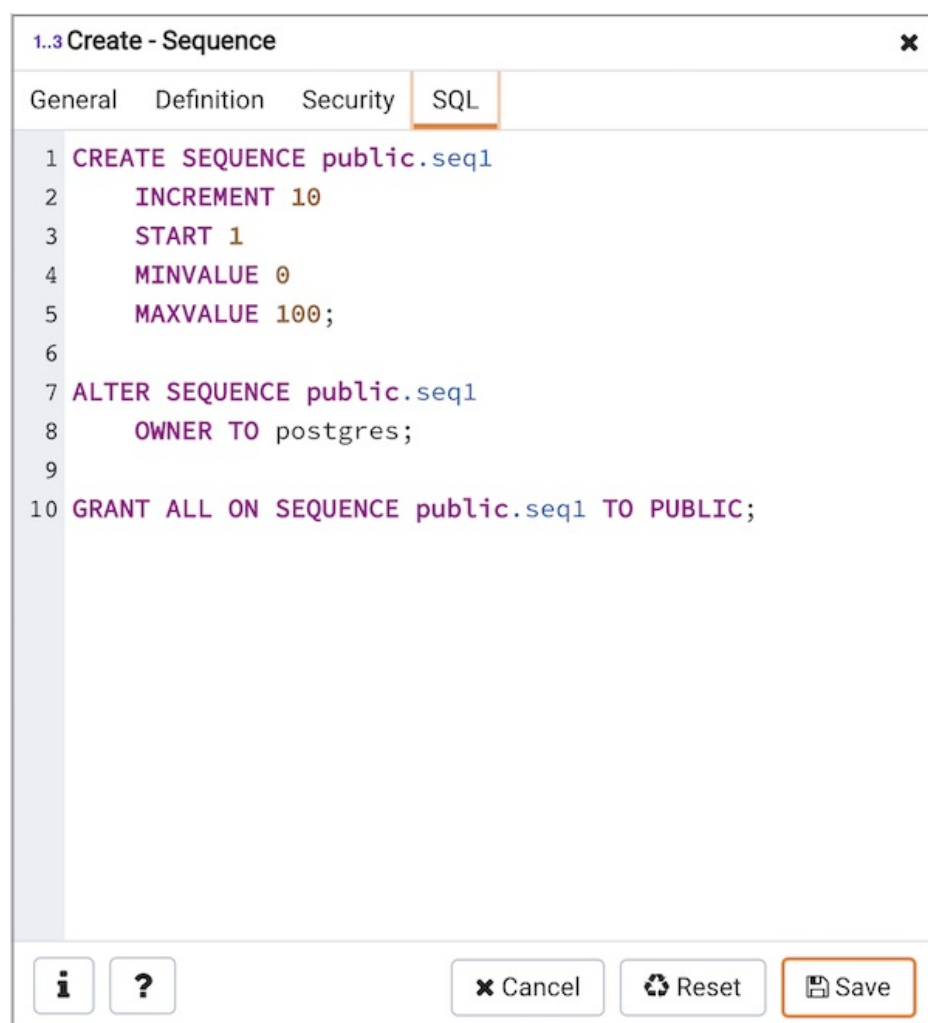
Your entries in the **Sequence** dialog generate a generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

Example

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

736

The following is an example of the sql command generated by user selections in the **Sequence** dialog:



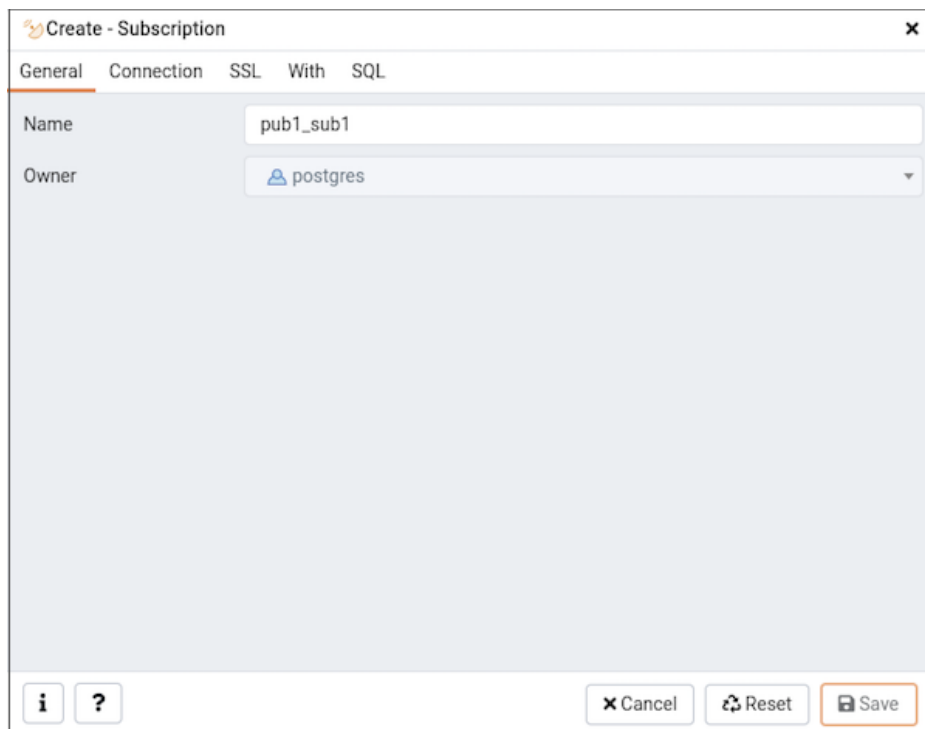
The example shown demonstrates a sequence named **seconds**. The sequence will increase in **5** second increments, and stop when it reaches a maximum value equal of **60**.

- Click the **Info** button (i) to access online help. View context-sensitive help in the **Tabbed browser**, where a new tab displays the PostgreSQL core documentation.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

### 34.5.11.22 Subscription Dialog

Use the **Subscription** dialog to create a subscription. A subscription defines the connection to another database and set of publications (one or more) to which it wants to subscribe.

The **subscription** dialog organizes the development of a subscription through the following dialog tabs: **General**, **Connection** and **With**. The **SQL** tab displays the SQL code generated by dialog selections.



**Create - Subscription**

General Connection SSL With SQL

Name: pub1\_sub1

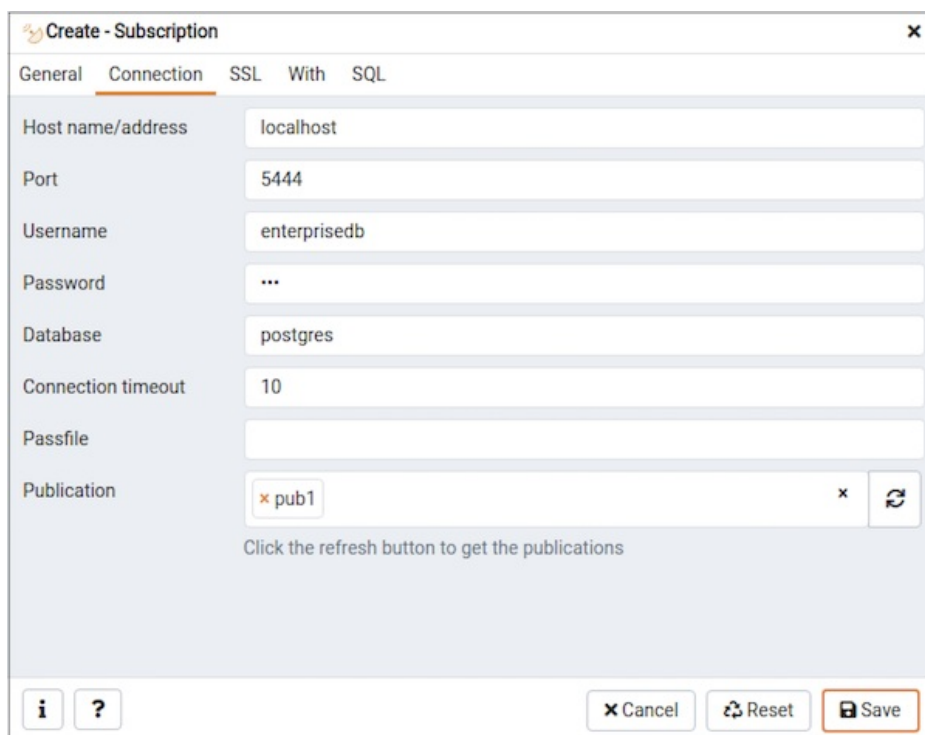
Owner: postgres

Buttons: [i] [?] [Cancel] [Reset] [Save]

Use the fields in the **General** tab to identify the subscription:

- Use the **Name** field to add a descriptive name for the subscription. The name will be displayed in the Browser tree control.
- The **Owner** field takes the name of the user automatically as current connected user. You can change the owner after creating subscription using alter subscription. Please note that owner of the subscription has superuser privileges.

Click the **Connection** tab to continue.



**Create - Subscription**

General Connection SSL With SQL

Host name/address: localhost

Port: 5444

Username: enterprisedb

Password: ...

Database: postgres

Connection timeout: 10

Passfile:

Publication: pub1 [x] [refresh]

Click the refresh button to get the publications

Buttons: [i] [?] [Cancel] [Reset] [Save]

Use the **Connection** tab to define the connection string to the publisher:

- Use the **Host name/address** field to provide the valid hostname or ip address of the publication.
- Use the **Port** field to provide port number to connect at Postgres Server in which publication is residing.



- Use the **Username** field to provide the name of the user to connect to the publication.
- Use the **Password** to provide the password of the user.
- Use the **Database** field to connect to the database in which publication is residing.
- Use the **Connection timeout** field to specify the maximum wait for connection, in seconds. Zero or not specified means wait indefinitely. It is not recommended to use a timeout of less than 2 seconds.
- Use the **Passfile** field to specify the location of a password file (.pgpass). A .pgpass file allows a user to login without providing a password when they connect. For more information, see [Section 33.15 of the Postgres documentation](#).
- Use the **Publication** field to specify the publication name on the publishers to subscribe to. Click on the refresh button at the end to load the names of the existing publications and then select from the list. You can also write the name of the known publication in the field.

Click the **SSL** tab to continue.

**Create - Subscription**

General Connection **SSL** With SQL

SSL mode: Prefer

Client certificate: [File Selection Icon]

Client certificate key: [File Selection Icon]

Root certificate: [File Selection Icon]

Certificate revocation list: [File Selection Icon]

SSL compression? [No]

[i] [?] [Cancel] [Reset] [Save]

Use the fields in the **SSL** tab to configure SSL:

- Use the drop-down list box in the **SSL** field to select the type of SSL connection the server should use. For more information about using SSL encryption, see [Section 33.18 of the Postgres documentation](#).

You can use the platform-specific File manager dialog to upload files that support SSL encryption to the server. To access the File manager dialog, click the icon that is located to the right of each of the following fields.

- Use the **Client certificate** field to specify the file containing the client SSL certificate. This file will replace the default `<STORAGE_DIR>/<USERNAME>/.postgresql/postgresql.crt`. This parameter is ignored if an SSL connection is not made.
- Use the **Client certificate key** field to specify the file containing the secret key used for the client certificate. This file will replace the default `<STORAGE_DIR>/<USERNAME>/.postgresql/postgresql.key`. This parameter is ignored if an SSL connection is not made.
- Use the **Root certificate** field to specify the file containing the SSL certificate authority. This file will replace the default `~/.postgresql/root.crt`. This parameter is ignored if an SSL connection is not made.
- Use the **Certificate revocation list** field to specify the file containing the SSL certificate revocation list. This list will replace the default list, found in `~/.postgresql/root.crl`. This parameter is ignored if an SSL connection is not made.
- When **SSL compression?** is set to **True**, data sent over SSL connections will be compressed. The default value is **False** (compression is disabled). This parameter is ignored if an SSL connection is not made.

#### Warning

The certificates, private keys, and the revocation list are stored in the per-user file storage area on the server, which is owned by the user account under which the PEM server process is run. This means that administrators of the server may be able to access those files; appropriate caution should be taken before choosing to use this feature.

Click the **With** tab to continue.

**Create - Subscription**

General Connection SSL **With** SQL

**Copy data?** ☒ Yes  
Specifies whether the existing data in the publications that are being subscribed to should be copied once the replication starts.

**Create slot?** ☐ No  
Specifies whether the command should create the replication slot on the publisher. This field will be disabled and set to false if subscription connects to same database. Otherwise, the CREATE SUBSCRIPTION call will hang.

**Enabled?** ☒ Yes  
Specifies whether the subscription should be actively replicating, or whether it should be just setup but not started yet.

**Connect?** ☒ Yes  
Specifies whether the CREATE SUBSCRIPTION should connect to the publisher at all. Setting this to false will change default values of enabled, create\_slot and copy\_data to false.

**Slot name**  
  
Name of the replication slot to use. The default behavior is to use the name of the subscription for the slot name.

**Synchronous commit**   
The value of this parameter overrides the synchronous\_commit setting. The default value is off.

**i ?** **Cancel** **Reset** **Save**

Use the **With** tab to define some parameters for a subscription:

- The **Copy data?** switch specifies whether the existing data in the publications that are being subscribed to should be copied once the replication starts. By default it is set to *Yes*.
- The **Create slot?** switch specifies whether the command should create the replication slot on the publisher. By default it is set to *Yes*. Please note: if your publisher and subscriber both are inside same PostgreSQL server then it is set to *No* by default.
- The **Enabled?** switch specifies whether the subscription should be actively replicating, or whether it should be just setup but not started yet. By default it is set to *Yes*.
- The **Connect?** specifies whether the CREATE SUBSCRIPTION should connect to the publisher at all. By default, it is set to *Yes*. Setting this to *No* will change default values of enabled, create\_slot and copy\_data to *No*.
- Use the **Slot Name** field to specify the name of the replication slot to use. By default, it uses the name of the subscription for the slot name.
- Use the **Synchronous commit** field to override the synchronous\_commit setting. By default, it is set to *off*. It is safe to use off for logical replication: If the subscriber loses transactions because of missing synchronization, the data will be sent again from the publisher.

Click the **SQL** tab to continue.

Your entries in the **Subscription** dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

### Example

The following is an example of the sql command generated by user selections in the **Subscription** dialog:

**Create - Subscription**

General Connection SSL With SQL

```

1 CREATE SUBSCRIPTION pub1_sub1
2 CONNECTION 'host=localhost port=5444 user=enterprisedb dbname=postgres connect_timeout=10 password=xxxxxx sslmode=prefer'
3 PUBLICATION pub1
4 WITH (connect = true, enabled = true, copy_data = true, create_slot = false, synchronous_commit = 'off');

```

**i ?** **Cancel** **Reset** **Save**

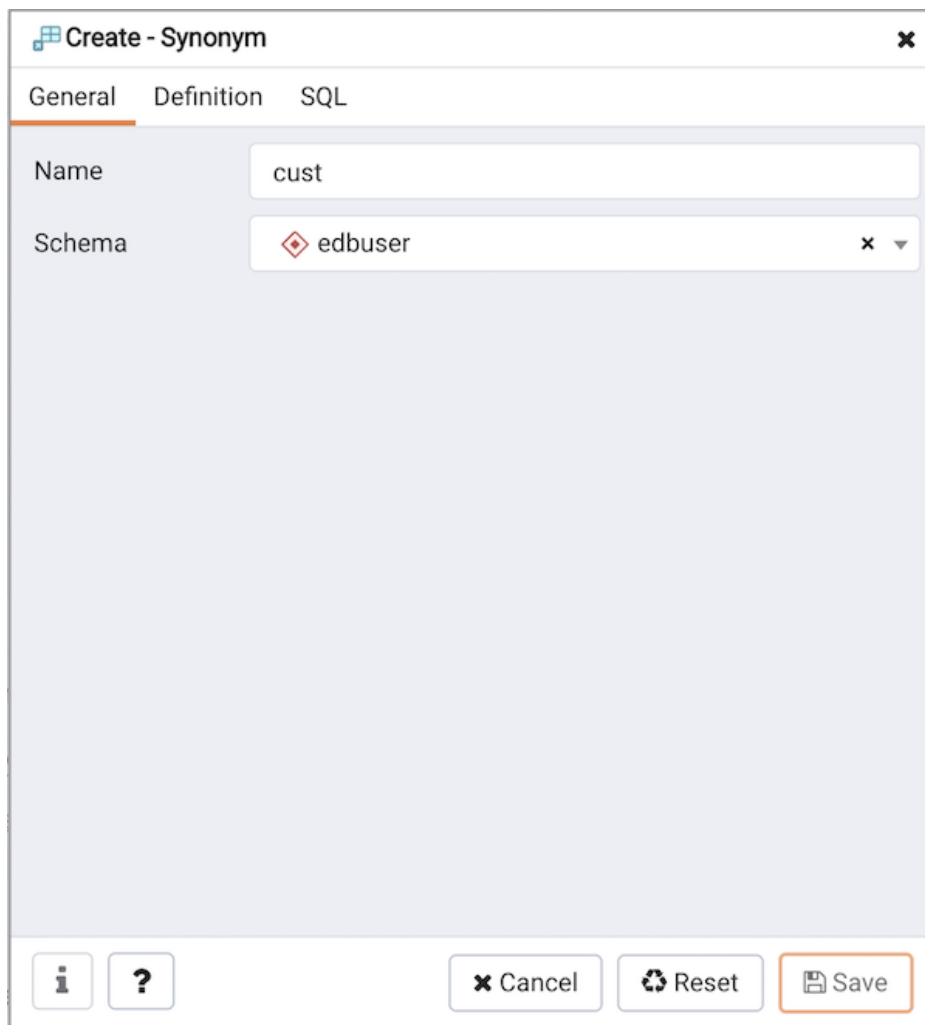
The example creates a subscription named `pub1_sub1` that is owned by `postgres`. It will replicate the data from the publication `pub1`.

- Click the **Info** button (i) to access online help.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore all the default settings.

### 34.5.11.23 Synonym Dialog

Use the **Synonym** dialog to substitute the name of a target object with a user-defined synonym.

The **Synonym** dialog organizes the development of a synonym through the **General** tab. The **SQL** tab displays the SQL code generated by dialog selections.

The image shows a 'Create - Synonym' dialog box with three tabs: 'General', 'Definition', and 'SQL'. The 'General' tab is selected. It contains two input fields: 'Name' with the value 'cust' and 'Schema' with a dropdown menu showing 'edbuser'. At the bottom, there are three buttons: 'Cancel', 'Reset', and 'Save' (which is highlighted with an orange border). There are also information and help icons on the left.

Create - Synonym

General Definition SQL

Name cust

Schema edbuser

Cancel Reset Save

Use the fields in the **General** tab to identify the synonym:

- Use the **Name** field to specify the name of synonym. The name will be displayed in the **Browser** tree control.
- Select the name of the schema in which the synonym will reside from the drop-down listbox in the **Schema** field.

In the definition panel, identify the target:

- Use the drop-down listbox next to **Target Type** to select the the type of object referenced by the synonym.
- Use the drop-down listbox next to **Target Schema** to select the name of the schema in which the object resides.
- Use the drop-down listbox next to **Target Object** to select the name of the object referenced by the synonym.

Click the **SQL** tab to continue.

Your selections and entries in the **Synonym** dialog generate a SQL command.

**Create - Synonym**

General Definition **SQL**

```

1 CREATE OR REPLACE SYNONYM edbuser.cust
2   FOR edbuser.customers;

```

Cancel Reset **Save**

The example creates a synonym for the `emp` table named `emp_hist`.

- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

### 34.5.11.24 Trigger function Dialog

Use the **Trigger function** dialog to create or manage a trigger\_function. A trigger function defines the action that will be invoked when a trigger fires.

The **Trigger function** dialog organizes the development of a trigger function through the following dialog tabs: **General**, **Definition**, **Code**, **Options**, **Parameters** and **Security**. The **SQL** tab displays the SQL code generated by dialog selections.

**Create - Trigger function**

General Definition Code Options Parameters Security SQL

Name: record\_update

Owner: enterisedb

Schema: public

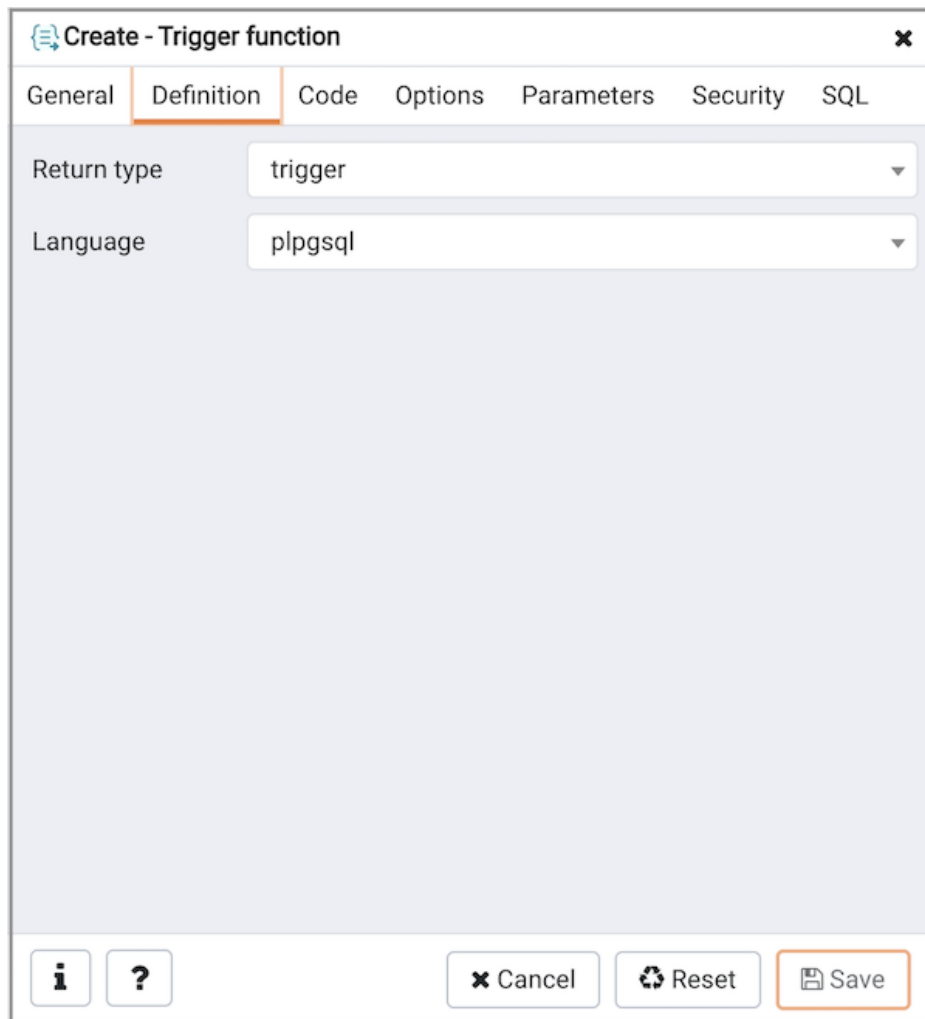
Comment:

Buttons: ? i Cancel Reset Save

Use the fields in the **General** tab to identify the trigger function:

- Use the **Name** field to add a descriptive name for the trigger function. The name will be displayed in the **Browser** tree control. Please note that trigger functions will be invoked in alphabetical order.
- Use the drop-down listbox next to **Owner** to select the role that will own the trigger function.
- Select the name of the schema in which the trigger function will reside from the drop-down listbox in the **Schema** field.
- Store notes about the trigger function in the **Comment** field.

Click the **Definition** tab to continue.



The screenshot shows a 'Create - Trigger function' dialog box with the following elements:

- Tabbed Interface:** The 'Definition' tab is selected, with other tabs including 'General', 'Code', 'Options', 'Parameters', 'Security', and 'SQL'.
- Return type:** A dropdown menu showing 'trigger'.
- Language:** A dropdown menu showing 'plpgsql'.
- Buttons:** At the bottom, there are buttons for 'Cancel', 'Reset', and 'Save' (highlighted with an orange border). There are also information and help icons on the left.

Use the fields in the **Definition** tab to define the trigger function:

- Use the drop-down listbox next to **Return type** to specify the pseudotype that is associated with the trigger function:
  - Select **trigger** if you are creating a DML trigger.
  - Select **event\_trigger** if you are creating a DDL trigger.
- Use the drop-down listbox next to **Language** to select the implementation language. The default is **plpgsql**.

Click the **Code** tab to continue.

Create - Trigger function

General

Definition

Code

Options

Parameters

Security

SQL

1

BEGIN

2

3

END;

i

?

Cancel

Reset

Save

- Use the **Code** field to write the code that will execute when the trigger function is called.

Click the **Options** tab to continue.



**Create - Trigger function**

General Definition Code **Options** Parameters Security SQL

Volatility: **STABLE**

Returns a set?: ☐ No

Strict?: ☐ No

Security of definer?: ☐ No

Window?: ☐ No

Estimated cost:

Estimated rows:

Leak proof?: ☐ No

**Save** | Cancel | Reset

Use the fields in the **Options** tab to describe or modify the action of the trigger function:

- Use the drop-down listbox next to **Volatility** to select one of the following:
  - **VOLATILE** indicates that the trigger function value can change even within a single table scan.
  - **STABLE** indicates that the trigger function cannot modify the database, and that within a single table scan it will consistently return the same result for the same argument values.
  - **IMMUTABLE** indicates that the trigger function cannot modify the database and always returns the same result when given the same argument values.
- Move the **Returns a Set?** switch to indicate if the trigger function returns a set that includes multiple rows. The default is **No**.
- Move the **Strict?** switch to indicate if the trigger function always returns NULL whenever any of its arguments are NULL. If **Yes**, the function is not executed when there are NULL arguments; instead a NULL result is assumed automatically. The default is **No**.
- Move the **Security of definer?** switch to specify that the trigger function is to be executed with the privileges of the user that created it. The default is **No**.
- Move the **Window?** switch to indicate that the trigger function is a window function rather than a plain function. The default is **No**. This is currently only useful for trigger functions written in C.
- Use the **Estimated cost** field to specify a positive number representing the estimated execution cost for the trigger function, in units of `cpu_operator_cost`. If the function returns a set, this is the cost per returned row.
- Use the **Estimated rows** field to specify a positive number giving the estimated number of rows that the query planner should expect the trigger function to return. This is only allowed when the function is declared to return a set. The default assumption is 1000 rows.

- Move the **Leak proof?** switch to indicate whether the trigger function has side effects. The default is **No** . This option can only be set by the superuser.

Click the **Parameters** tab to continue.

Create - Trigger function

General

Definition

Code

Options

Parameters

Security

SQL

|  | Name        | Value            |
|--|-------------|------------------|
|  | array_nulls | <div>False</div> |

i

?

Cancel

Reset

Save

Use the fields in the **Parameters** tab to specify settings that will be applied when the trigger function is invoked. Click the **Add** icon (+) to add a **Name** / **Value** pair to the table below.

- Use the drop-down listbox in the **Name** field to select a parameter.
- Use the **Value** field to specify the value that will be associated with the selected parameter. This field is context-sensitive.

Click the **Add** icon (+) to set additional parameters; to discard a parameter, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Click the **Security** tab to continue.

Create - Trigger function

General

Definition

Code

Options

Parameters

Security

SQL

Privileges

Grantee

Privileges

Grantor

Security labels

Provider

Security label

i

?

Cancel

Reset

Save

Use the **Security** tab to assign privileges and define security labels.

Use the **Privileges** panel to assign usage privileges for the trigger function to a role. Click the **Add** icon (+) to add a role to the table.

- Select the name of the role from the drop-down listbox in the **Grantee** field.
- Click inside the **Privileges** field. Check the boxes to the left of one or more privileges to grant the selected privilege to the specified user.
- The current user, who is the default grantor for granting the privilege, is displayed in the **Grantor** field.

Click the **Add** icon (+) to assign additional privileges; to discard a privilege, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Use the **Security Labels** panel to define security labels applied to the trigger function. Click the **Add** icon (+) to add each security label selection:

- Specify a security label provider in the **Provider** field. The named provider must be loaded and must consent to the proposed labeling operation.
- Specify a security label in the **Security Label** field. The meaning of a given label is at the discretion of the label provider. PostgreSQL places no restrictions on whether or how a label provider must interpret security labels; it merely provides a mechanism for storing them.

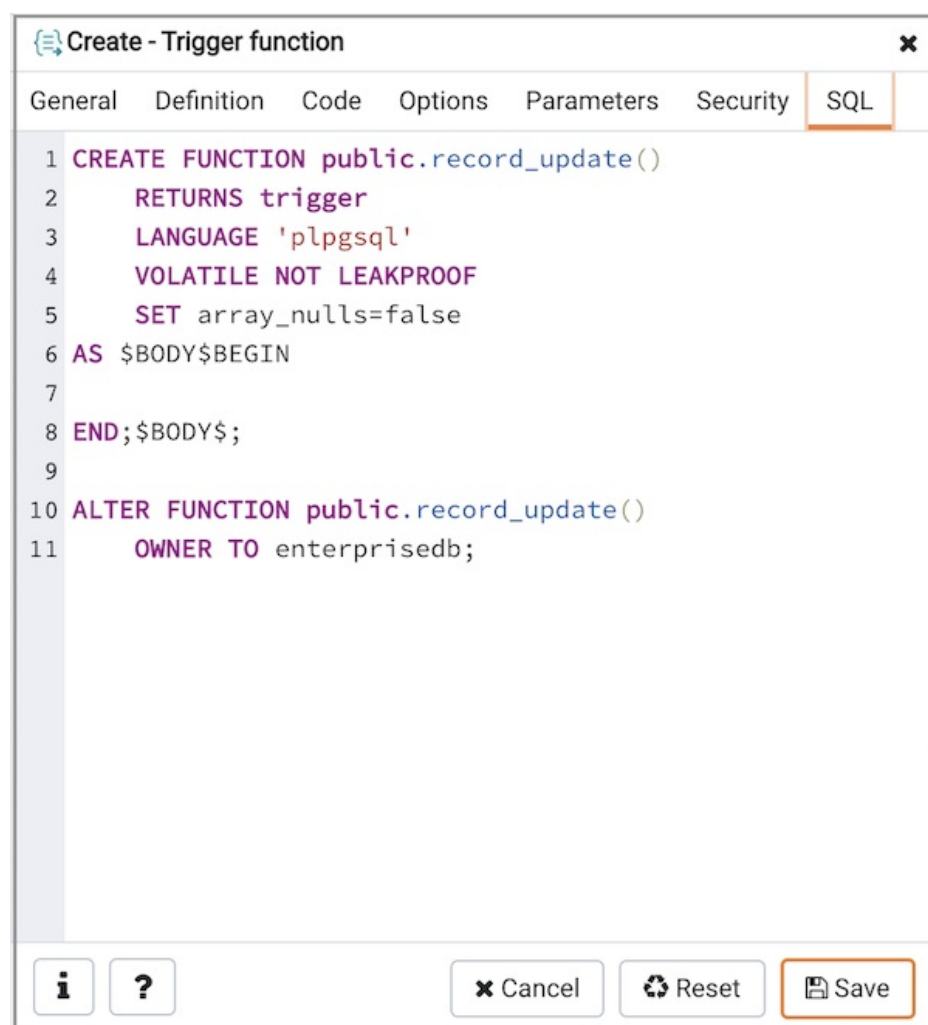
Click the **Add** icon (+) to assign additional security labels; to discard a security label, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Click the **SQL** tab to continue.

Your entries in the **Trigger function** dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit other tabs to modify the SQL command.

Example

The following is an example of the sql command generated by user selections in the **Trigger function** dialog:



The example shown demonstrates creating a trigger function named **emp\_stamp** that checks for a new employee's name, and checks that the employee's salary is a positive value.

- Click the **Info** button (i) to access online help. View context-sensitive help in the **Tabbed browser**, where a new tab displays the PostgreSQL core documentation.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

### 34.5.11.25 Type Dialog

Use the **Type** dialog to register a custom data type.

The **Type** dialog organizes the development of a data type through the following dialog tabs: **General**, **Definition**, and **Security**. The **SQL** tab displays the SQL code generated by dialog selections.

**Create - Type**

General Definition Security SQL

Name: work\_order

Owner: postgres

Schema: public

Comment:

Buttons: [i] [?] [x Cancel] [Reset] [Save]

Use the fields in the **General** tab to identify the custom data type:

- Use the **Name** field to add a descriptive name for the type. The name will be displayed in the **Browser** tree control. The type name must be distinct from the name of any existing type, domain, or table in the same schema.
- Use the drop-down listbox next to **Owner** to select the role that will own the type.
- Select the name of the schema in which the type will reside from the drop-down listbox in the **Schema** field.
- Store notes about the type in the **Comments** field.

Click the **Definition** tab to continue.

Select a data type from the drop-down listbox next to **Type** on the **Definition** tab; the panel below changes to display the options appropriate for the selected data type. Use the fields in the panel to define the data type.

There are five data types:

- *Composite Type*
- *Enumeration Type*
- *Range Type*
- **External Type** (or **Base Type**)
- *Shell Type*

If you select **Composite** in the **Type** field, the **Definition** tab displays the **Composite Type** panel:

Create - Type

General

Definition

Security

SQL

Type

Composite

Composite Type

+

|  | Member Name | Type     | Length/Precision | Scale | Collation |
|--|-------------|----------|------------------|-------|-----------|
|  | data_member | "char"[] |                  |       |           |

i

?

Cancel

Reset

Save

Click the **Add** icon (+) to provide attributes of the type. Fields on the **General** panel are context sensitive and may be disabled.

- Use the **Member Name** field to add an attribute name.
- Use the drop-down listbox in the **Type** field to select a datatype.
- Use the **Length/Precision** field to specify the maximum length of a non-numeric type, or the total count of significant digits in a numeric type.
- Use the **Scale** field to specify the number of digits to the right of the decimal point.
- Use the drop-down listbox in the **Collation** field to select a collation (if applicable).

Click the **Add** icon (+) to define an additional member; click the trash icon to the left of the row to discard a row.

If you select the **Enumeration** in the **Type** field, the **Definition** tab displays the **Enumeration Type** panel:

Create - Type

General

Definition

Security

SQL

Type

Enumeration

Enumeration type

+

Label

data\_label

i

?

✕ Cancel

↺ Reset

💾 Save

Click the **Add** icon (+) to provide a label for the type.

- Use the **Label** field to add a label, which must be less than 64 bytes long.

Click the **Add** icon (+) after each selection to create additional labels; to discard a label, click the trash icon to the left of the row.

If you select **External**, the **Definition** tab displays the **External Type** panel:

**Create - Type**

General **Definition** Security SQL

Type: External

Required Optional-1 Optional-2

Input function: pg\_catalog.domain\_in

Output function: pg\_catalog.numeric\_sqrt

Buttons: [i] [?] [x Cancel] [Reset] [Save]

On the **Required** tab:

- Use the drop-down listbox next to the **Input function** field to add an input\_function. The input\_function converts the type's external textual representation to the internal representation used by the operators and functions defined for the type.
- Use the drop-down listbox next to the **Output function** field to add an output\_function. The output\_function converts the type's internal representation used by the operators and functions defined for the type to the type's external textual representation.

On the **Optional-1** tab:

- Use the drop-down listbox next to the optional **Receive function** field to select a receive\_function. The optional receive\_function converts the type's external binary representation to the internal representation. If this function is not supplied, the type cannot participate in binary input.
- Use the drop-down listbox next to the optional **Send function** field to select a send\_function. The optional send\_function converts from the internal representation to the external binary representation. If this function is not supplied, the type cannot participate in binary output.
- Use the drop-down listbox next to the optional **Typmod in function** field tab to select a type\_modifier\_input\_function.
- Use the drop-down listbox next to the optional **Typmod out function** field tab to select a type\_modifier\_output\_function. It is allowed to omit the type\_modifier\_output\_function, in which case the default display format is the stored typmod integer value enclosed in parentheses.
- Use the optional **Internal length** to specify a value for internal representation.
- Move the **Variable?** switch to specify the internal representation is of variable length (VARIABLE). The default is a fixed length positive integer.
- Specify a default value in the optional **Default** field in cases where a column of the data type defaults to something other than the null value. Specify the default with the DEFAULT key word. (A default can be overridden by an explicit DEFAULT clause attached to a particular column.)
- Use the drop-down listbox next to the optional **Analyze function** field to select a function for performing type-specific statistics collection for columns of the data type.
- Use the drop-down listbox next to the optional **Category type** field to help control which implicit cast will be applied in ambiguous situations.
- Move the **Preferred?** switch to **Yes** to specify the selected category type is preferred. The default is **No**.

On the **Optional-2** tab:



- Use the drop-down listbox next to the optional **Element type** field to specify a data type.
- Use the optional **Delimiter** field to indicate the delimiter to be used between values in the external representation of arrays for this data type. The default delimiter is the comma (.). Note that the delimiter is associated with the array element type, not the array type itself.
- Use the drop-down listbox next to **Alignment type** to specify the storage alignment required for the data type. The allowed values (char, int2, int4, and double) correspond with alignment on 1, 2, 4, or 8 byte boundaries.
- Use the drop-down listbox next to optional **Storage type** to select a strategy for storing data.
- Move the **Passed by value?** switch to **Yes** to override the existing data type value. The default is **No**.
- Move the **Collatable?** switch to **Yes** to specify column definitions and expressions of the type may carry collation information through use of the COLLATE clause. The default is **No**.

If you select **Range** in the **Type** field, the **Definition** tab displays the **Range** panel. Fields on the **Range** panel are context-sensitive and may be disabled.

Create - Type

General

Definition

Security

SQL

Type

Range

Subtype

anyrange

Subtype operator class

Collation

Canonical function

Subtype diff function

i

?

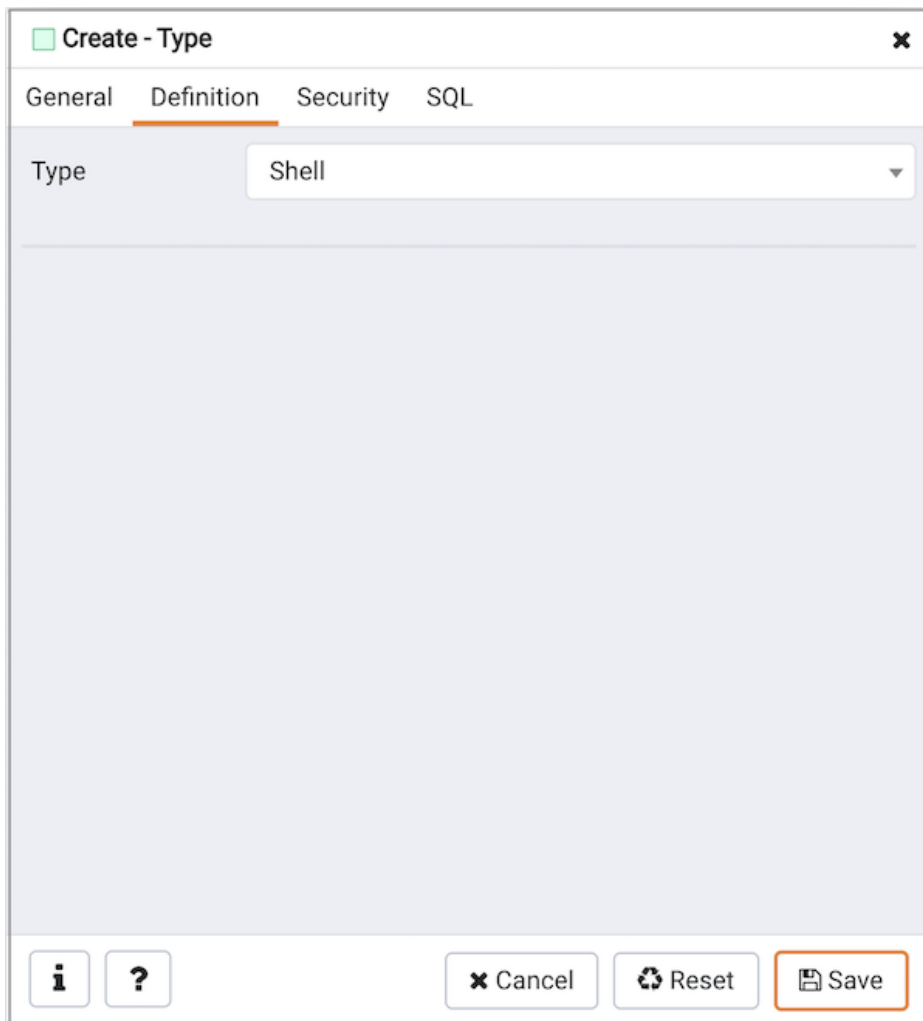
Cancel

Reset

Save

- Use the drop-down listbox next to **Sub-type** to select an associated b-tree operator class (to determine the ordering of values for the range type).
- Use the drop-down listbox next to **Sub-type operator class** to use a non-default operator class.
- Use the drop-down listbox next to **Collation** to use a non-default collation in the range's ordering if the sub-type is collatable.
- Use the drop-down listbox next to **Canonical function** to convert range values to a canonical form.
- Use the drop-down listbox next to **Sub-type diff function** to select a user-defined subtype\_diff function.

If you select **Shell** in the **Type** field, the **Definition** tab displays the **Shell** panel:



The image shows a 'Create - Type' dialog box with a title bar containing a green square icon and a close button. Below the title bar are four tabs: 'General', 'Definition', 'Security', and 'SQL'. The 'Definition' tab is selected and highlighted with an orange underline. Inside the dialog, there is a 'Type' label and a dropdown menu showing 'Shell'. The main area of the dialog is a large, empty light blue rectangle. At the bottom, there are five buttons: an information icon (i), a question mark icon (?), a 'Cancel' button with a close icon (x), a 'Reset' button with a refresh icon, and a 'Save' button with a floppy disk icon. The 'Save' button is highlighted with an orange border.

A shell type is a placeholder for a type and has no parameters.

Click the **Security** tab to continue.

Use the **Privileges** panel to assign privileges for the type; click the **Add** icon (+) to grant privileges:

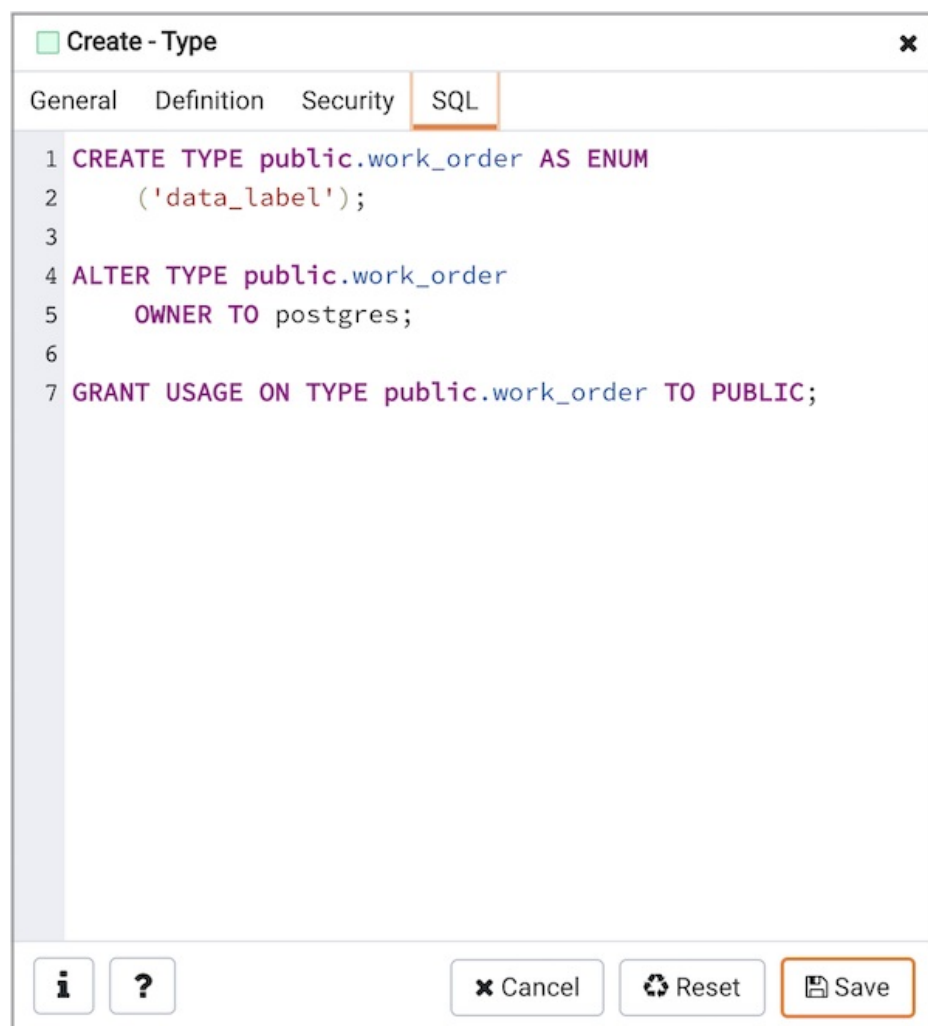
- Click the **Add** icon (+) to assign additional privileges; to discard a privilege, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

- Specify a security label provider in the **Provider** field. The named provider must be loaded and must consent to the proposed labeling operation.
- Specify a security label in the **Security Label** field. The meaning of a given label is at the discretion of the label provider. PostgreSQL places no restrictions on whether or how a label provider must interpret security labels; it merely provides a mechanism for storing them.

Click the **SQL** tab to continue.

### Example

The following is an example of a sql command generated by user selections made in the **Type** dialog:



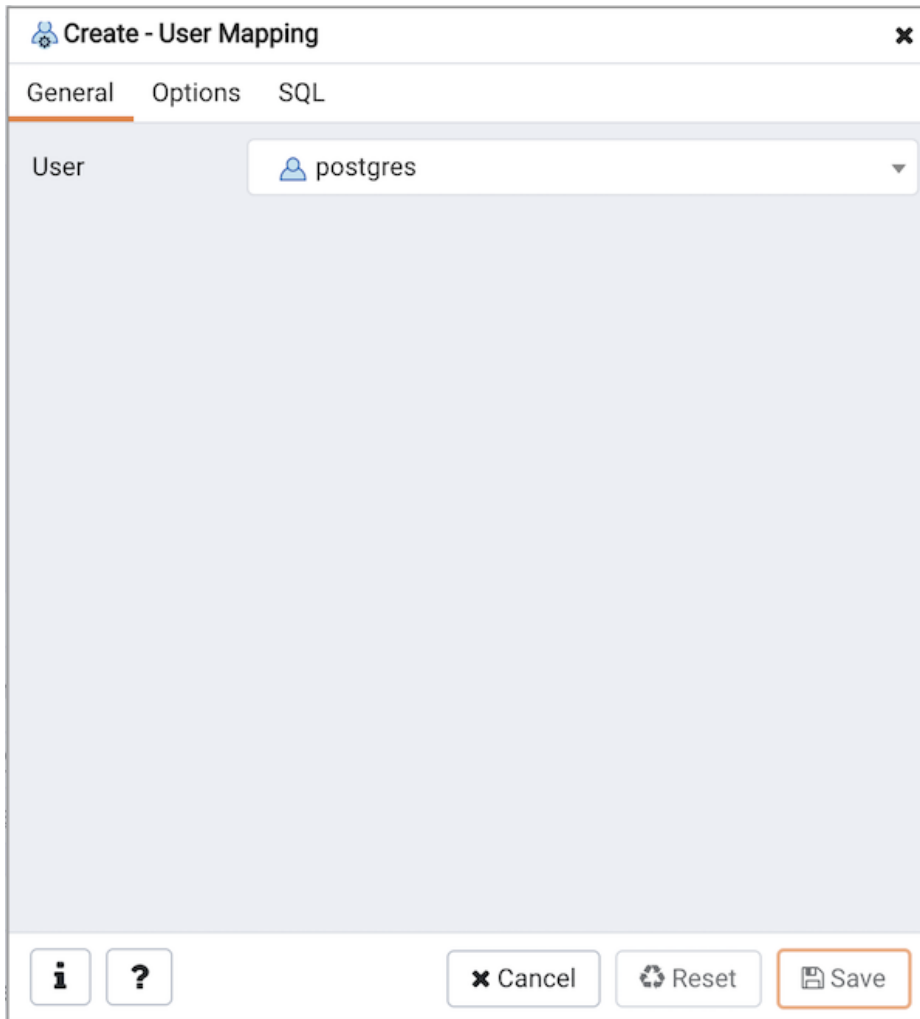
The example shown demonstrates creating a data type named **work\_order**. The data type is an enumerated type with three labels: new, open and closed.

- Click the **Info** button (i) to access online help. View context-sensitive help in the **Tabbed browser**, where a new tab displays the PostgreSQL core documentation.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

### 34.5.11.26 User Mapping Dialog

Use the **User Mapping** dialog to define a new mapping of a user to a foreign server.

The **User Mapping** dialog organizes the development of a user mapping through the following dialog tabs: **General** and **Options**. The **SQL** tab displays the SQL code generated by dialog selections.



**Create - User Mapping**

General Options SQL

User postgres

**i ?** **Cancel** **Reset** **Save**

Use the drop-down listbox in the **User** field in the **General** tab to identify the connecting role:

- Select **CURRENT\_USER** to use the name of the current role.
- Select **PUBLIC** if no other user-specific mapping is applicable.
- Select a pre-defined role name to specify the name of an existing user.

Click the **Options** tab to continue.

Create - User Mapping

General

Options

SQL

Options

|  | Options  | Value    |
|--|----------|----------|
|  | user     | postgres |
|  | password | postgres |

Cancel

Reset

Save

Use the fields in the **Options** tab to specify connection options; the accepted option names and values are specific to the foreign data wrapper associated with the server specified in the user mapping. Click the **Add** button to add an option/value pair.

- Specify the option name in the **Option** field.
- Provide a corresponding value in the **Value** field.

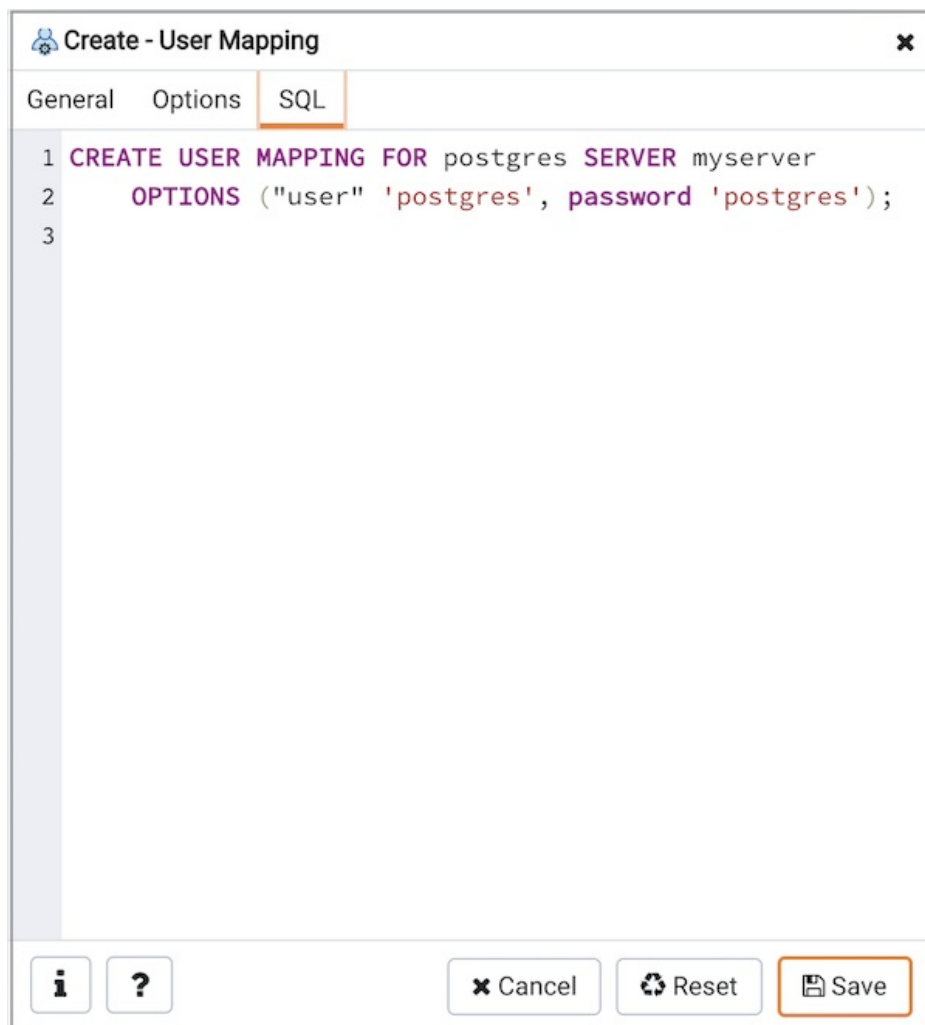
Click **Add** to specify each additional option/value pair; to discard an option, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Click the **SQL** tab to continue.

Your entries in the **User Mapping** dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

Example

The following is an example of the sql command generated by user selections in the **User Mapping** dialog:



The example shown demonstrates a user mapping for the `hdfs_server`. The user is `CURRENT_USER` with a password `secret`.

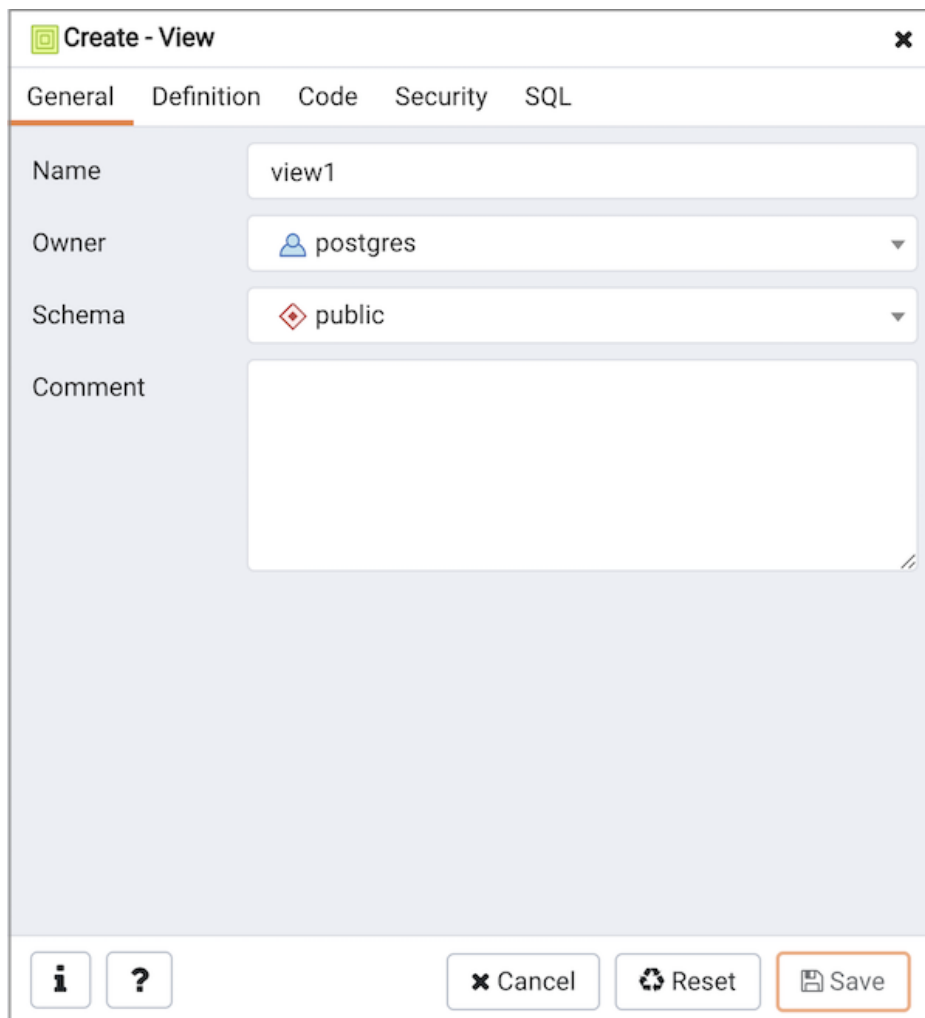
- Click the **Info** button (i) to access online help. View context-sensitive help in the **Tabbed browser**, where a new tab displays the PostgreSQL core documentation.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

### 34.5.11.27 View Dialog

Use the **View** dialog to define a view. The view is not physically materialized; the query is executed each time the view is referenced in a query.

The **View** dialog organizes the development of a View through the following dialog tabs: **General**, *Definition*, **Code** and *Security*". The **SQL** tab displays the SQL code generated by dialog selections.

Click the **General** tab to begin.



The image shows a 'Create - View' dialog box with a close button (X) in the top right corner. It has five tabs: 'General' (selected), 'Definition', 'Code', 'Security', and 'SQL'. The 'General' tab contains the following fields:

- Name:** A text input field containing 'view1'.
- Owner:** A dropdown menu with a person icon and the text 'postgres'.
- Schema:** A dropdown menu with a diamond icon and the text 'public'.
- Comment:** A large, empty text area.

At the bottom of the dialog, there are five buttons: an information icon (i), a question mark icon (?), a 'Cancel' button with an X icon, a 'Reset' button with a circular arrow icon, and a 'Save' button with a floppy disk icon.

Use the fields in the **General** tab to identify a view:

- Use the **Name** field to add a descriptive name for the view. The name of the view must be distinct from the name of any other view, table, sequence, index or foreign table in the same schema. The name will be displayed in the **Browser** tree control.
- Use the drop-down listbox next to **Owner** to select the role that will own the view.
- If applicable, select the name of the schema in which the view will reside from the drop-down listbox in the **Schema** field.
- Store notes about the view in the **Comments** field.

Click the **Definition** tab to continue.



**Create - View**

General **Definition** Code Security SQL

Security barrier? ☐ No

Check options Local ▼

**i** **?** **Cancel** **Reset** **Save**

Use the fields in the **Definition** tab to define properties of the view:

- Set the **Security Barrier** switch to **Yes** to indicate that the view is to act as a security barrier. For more information about defining and using a security barrier rule, see Section 38.5 of the PostgreSQL documentation.
- Use the drop-down listbox next to **Check options** to select from **No**, **Local** or **Cascaded**:
  - The **Local** option specifies that new rows are only checked against the conditions defined in the view. Any conditions defined on underlying base views are not checked (unless you specify the CHECK OPTION).
  - The **Cascaded** option specifies new rows are checked against the conditions of the view and all underlying base views.

Click the **Code** tab to continue.

Create - View

General

Definition

Code

Security

SQL

1

SELECT \* FROM public.try;

i

?

Cancel

Reset

Save

Use the workspace in the **Code** tab to write a query to create a view.

Click the **Security** tab to continue.

Create - View

General

Definition

Code

Security

SQL

Privileges

Grantee

Privileges

Grantor

PUBLIC

☐ ALL

☐ WITH GRANT OPTION

☐ INSERT

☐ WITH GRANT OPTION

☒ SELECT

☐ WITH GRANT OPTION

☐ UPDATE

☐ WITH GRANT OPTION

☐ DELETE

☐ WITH GRANT OPTION

☐ TRUNCATE

☐ WITH GRANT OPTION

☐ REFERENCES

☐ WITH GRANT OPTION

☐ TRIGGER

☐ WITH GRANT OPTION

postgres

Security labels

Provider

Security label

?

Cancel

Reset

Save

Use the **Security** tab to assign privileges and define security labels.

Use the **Privileges** panel to assign privileges to a role. Click the **Add** icon (+) to set privileges for the view:

- Select the name of the role that will be granted privileges from the drop-down listbox in the **Grantee** field.
- Click inside the **Privileges** field. Check the boxes to the left of one or more privileges to grant the selected privilege to the specified user.
- The current user, who is the default grantor for granting the privilege, is displayed in the **Grantor** field.

Click the **Add** icon (+) to assign additional privileges; to discard a privilege, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Use the **Security Labels** panel to define security labels applied to the view. Click the **Add** icon (+) to add each security label selection:

- Specify a security label provider in the **Provider** field. The named provider must be loaded and must consent to the proposed labeling operation.
- Specify a security label in the **Security Label** field. The meaning of a given label is at the discretion of the label provider. PostgreSQL places no restrictions on whether or how a label provider must interpret security labels; it merely provides a mechanism for storing them.

Click the **Add** icon (+) to assign additional security labels; to discard a security label, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Click the **SQL** tab to continue.

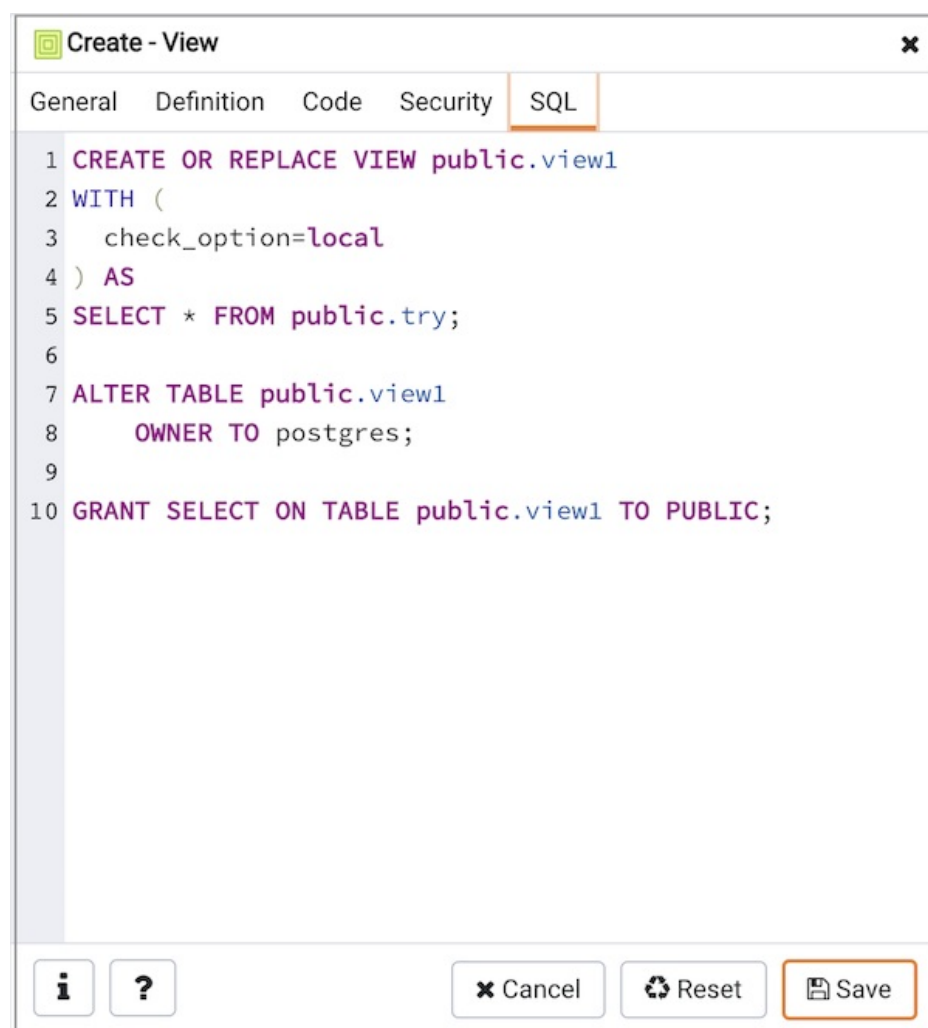
Your entries in the **View** dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

Example

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

765

The following is an example of the sql command generated by user selections in the **View** dialog:



The example shown demonstrates creating a view named **distributor\_codes** that includes the content of the **code** column from the **distributors** table.

- Click the **Info** button (i) to access online help. View context-sensitive help in the **Tabbed browser**, where a new tab displays the PostgreSQL core documentation.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

## 34.5.12 Creating or Modifying a Table

PEM provides dialogs that allow you to modify all table properties and attributes.

To access a dialog that allows you to create a database object, right-click on the object type in the **Browser** tree control, and select the **Create** option for that object. For example, to create a new table, Select a database from the tree control, select the schema under the database, right-click on the **Tables** node, and select *Create Table...*

Contents:

### 34.5.12.1 Check Dialog

Use the **Check** dialog to define or modify a check constraint. A check constraint specifies an expression that produces a Boolean result that new or updated rows must satisfy for an insert or update operation to succeed.

The **Check** dialog organizes the development of a check constraint through the **General** and **Definition** tabs. The **SQL** tab displays the SQL code generated by dialog selections.

✓ Create - Check

General Definition SQL

Name try\_chk

Comment

ⓘ ? × Cancel ↺ Reset 💾 Save

Use the fields in the **General** tab to identify the check constraint:

- Use the **Name** field to provide a descriptive name for the check constraint that will be displayed in the **Browser** tree control. With PostgreSQL 9.5 forward, when a table has multiple check constraints, they will be tested for each row in alphabetical order by name and after NOT NULL constraints.
- Store notes about the check constraint in the **Comment** field.

Click the **Definition** tab to continue.

✓ Create - Check

General Definition SQL

Check "No" > 0

No inherit? No

Don't validate? Yes

ⓘ ? × Cancel ↺ Reset 💾 Save

Use the fields in the **Definition** tab to define the check constraint:

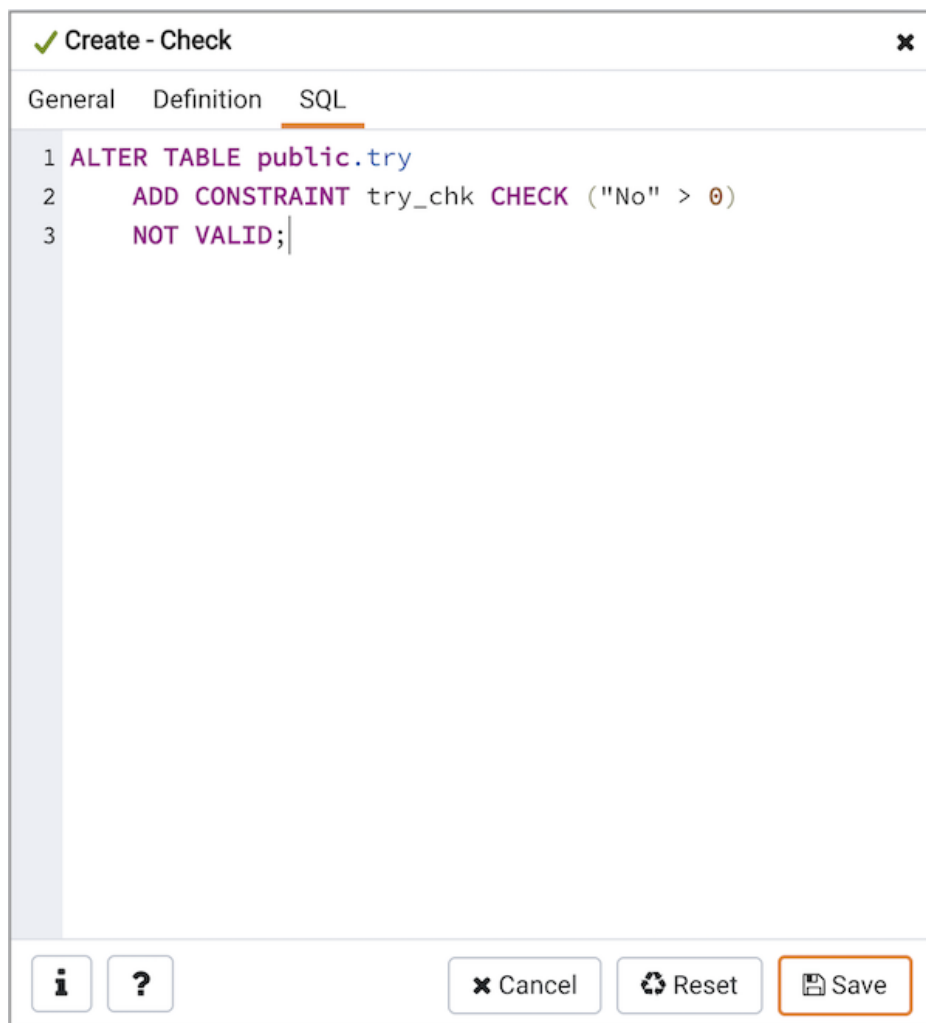
- Provide the expression that a row must satisfy in the **Check** field.
- Move the **No Inherit?** switch to the **Yes** position to specify this constraint is automatically inherited by a table's children. The default is **No**.
- Move the **Don't validate?** switch to the **No** position to skip validation of existing data; the constraint may not hold for all rows in the table. The default is **Yes**.

Click the **SQL** tab to continue.

Your entries in the **Check** dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

#### Example

The following is an example of the sql command generated by user selections in the **Check** dialog:



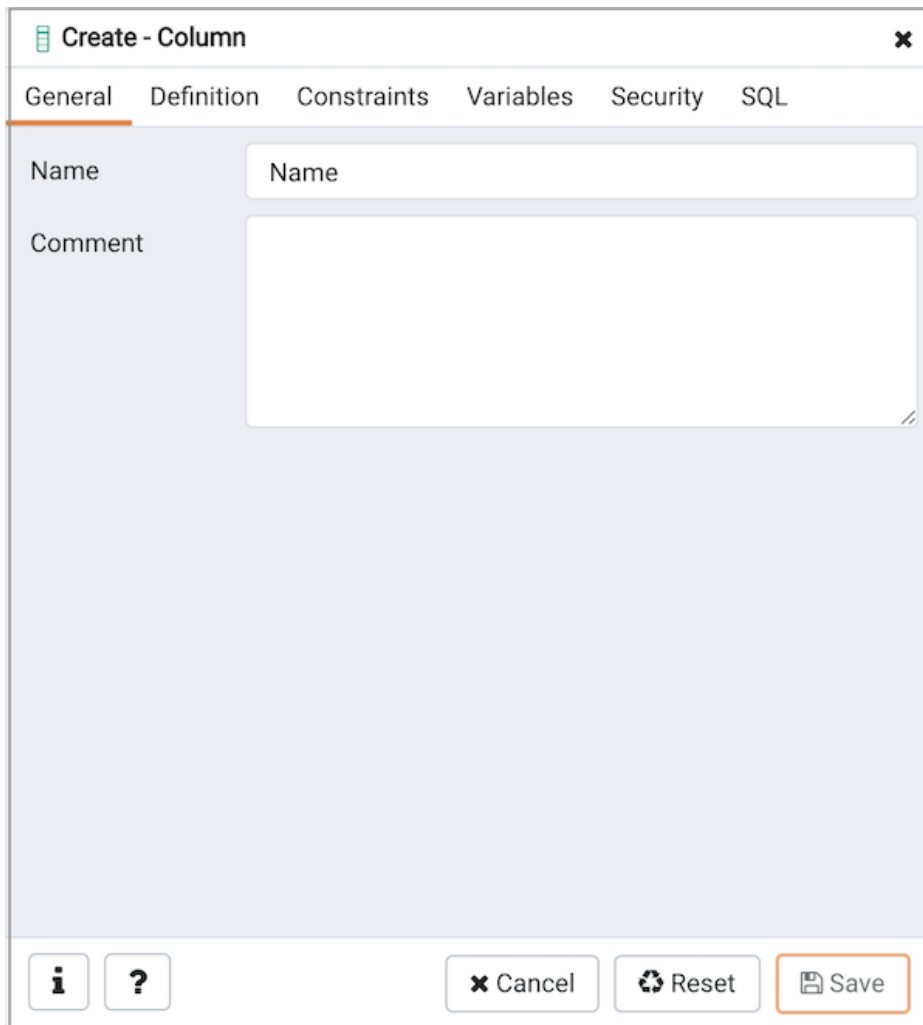
The example shown demonstrates creating a check constraint named `check_price` on the `price` column of the `products` table. The constraint confirms that any values added to the column are greater than 0.

- Click the **Info** button (i) to access online help. View context-sensitive help in the **Tabbed browser**, where a new tab displays the PostgreSQL core documentation.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

### 34.5.12.2 Column Dialog

Use the **Column** dialog to add a column to an existing table or modify a column definition.

The **Column** dialog organizes the development of a column through the following dialog tabs: **General**, **Definition**, and **Security**. The **SQL** tab displays the SQL code generated by dialog selections.



The image shows a 'Create - Column' dialog box with a title bar and a close button. It has five tabs: 'General', 'Definition', 'Constraints', 'Variables', and 'Security', with 'General' selected. The 'General' tab contains a 'Name' text field with the placeholder text 'Name' and a larger 'Comment' text area. At the bottom, there are four buttons: an information icon, a help icon, a 'Cancel' button, a 'Reset' button, and a 'Save' button.

Use the fields in the **General** tab to identify the column:

- Use the **Name** field to add a descriptive name for the column. The name will be displayed in the **Browser** tree control. This field is required.
- Store notes about the column in the **Comment** field.

Click the **Definition** tab to continue.



The screenshot shows a 'Create - Column' dialog box with the following elements:

- Header:** 'Create - Column' with a close button (X).
- Tabs:** 'General', 'Definition' (selected), 'Constraints', 'Variables', 'Security', 'SQL'.
- Fields:**
  - Data type:** A dropdown menu showing 'character varying'.
  - Length/Precision:** A text input field.
  - Scale:** A text input field.
  - Collation:** A dropdown menu showing 'Select from the list'.
- Footer:** Information icon (i), Help icon (?), 'Cancel' button, 'Reset' button, and 'Save' button (highlighted with an orange border).

Use the fields in the **Definition** tab to add parameters for the column. (Fields are disabled if inapplicable.)

- Use the drop-down listbox next to **Data Type** to select a data type for the column. For more information on the data types that are supported by PostgreSQL, refer to Chapter 8 of the Postgres core documentation. This field is required.
- Use the **Length/Precision** and **Scale** fields to specify the maximum number of significant digits in a numeric value, or the maximum number of characters in a text value.
- Use the drop-down listbox next to **Collation** to apply a collation setting to the column.

Click the **Constraints** tab to continue.

Create - Column

General

Definition

Constraints

Variables

Security

SQL

Default

Not NULL?

No

Type

✓ NONE

IDENTITY

GENERATED

i

?

Cancel

Reset

Save

Use the fields in the **Constraints** tab to specify constraints for the column. (Fields are disabled if inapplicable.)

- Use the **Default Value** field to specify a default data value.
- Move the **Not Null** switch to the **Yes** position to specify the column may not contain null values. The default is **No**.
- Use the **Type** field to specify the column type (NONE/IDENTITY/GENERATED). The default is **NONE**.

Click the **IDENTITY** type to create Identity column.

The screenshot shows the 'Create - Column' dialog box with the 'Constraints' tab selected. The 'Type' dropdown is set to 'IDENTITY' (highlighted with an orange border). The 'Identity' dropdown is set to 'ALWAYS'. The 'Cycled' toggle switch is set to 'No'. Other fields like 'Default', 'Not NULL?', 'Increment', 'Start', 'Minimum', 'Maximum', and 'Cache' are empty. At the bottom, there are buttons for 'Cancel', 'Reset', and 'Save' (highlighted with an orange border), along with information and help icons.

Use the following fields to create **IDENTITY** column. Identity columns are applicable for PG/EPAS version 10 and above.

- Use the **Identity** field to specify ALWAYS or BY DEFAULT. This clause is used to determine how the sequence value is given precedence over a user-specified value in an INSERT statement.
- Use the **Increment** field to specify which value is added to the current sequence value to create a new value.
- Provide a value in the **Start** field to specify the beginning value of the sequence. The default starting value is MINVALUE for ascending sequences and MAXVALUE for descending ones.
- Provide a value in the **Minimum** field to specify the minimum value a sequence can generate. If this clause is not supplied or NO MINVALUE is specified, then defaults will be used. The defaults are 1 and -263-1 for ascending and descending sequences, respectively.
- Provide a value in the **Maximum** field to specify the maximum value for the sequence. If this clause is not supplied or NO MAXVALUE is specified, then default values will be used. The defaults are 263-1 and -1 for ascending and descending sequences, respectively.
- Provide a value in the **Cache** field to specify how many sequence numbers are to be preallocated and stored in memory for faster access. The minimum value is 1 (only one value can be generated at a time, i.e., no cache), and this is also the default.
- Move the **Cycled** switch to the **Yes** position to allow the sequence to wrap around when the MAXVALUE or the MINVALUE has been reached by an ascending or descending sequence respectively. If the limit is reached, the next number generated will be the MINVALUE or MAXVALUE, respectively. The default is **No**.

Click the **GENERATED** type to create Generated column.

Create - Column

General

Definition

Constraints

Variables

Security

SQL

Default

Not NULL?

Yes

Type

NONE

IDENTITY

✓ GENERATED

Expression

i

?

✕ Cancel

🔄 Reset

💾 Save

Use the following fields to create **GENERATED** column. Generated columns are applicable for PG/EPAS version 12 and above.

- Use the **Expression** field to specify the generation expression. It can refer to other columns in the table, but not other generated columns. Any functions and operators used must be immutable. References to other tables are not allowed.

Click the **Variables** tab to continue.

**Create - Column**

General Definition Constraints **Variables** Security SQL

Variables +

| Name | Value |
|------|-------|
|------|-------|

ⓘ ? 
 ✕ Cancel
🔄 Reset
💾 Save

Use the **Variables** tab to specify the number of distinct values that may be present in the column; this value overrides estimates made by the ANALYZE command. Click the **Add** icon (+) to add a **Name** / **Value** pair:

- Select the name of the variable from the drop-down listbox in the **Name** field.
  - Select **n\_distinct** to specify the number of distinct values for the column.
  - Select **n\_distinct\_inherited** to specify the number of distinct values for the table and its children.
- Specify the number of distinct values in the **Value** field. For more information, see the documentation for [ALTER TABLE](#).

Click the **Add** icon (+) to specify each additional **Name** / **Value** pair; to discard a variable, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Click the **Security** tab to continue.

Create - Column

General

Definition

Constraints

Variables

Security

SQL

Security labels

| Provider | Security label |
|----------|----------------|
|----------|----------------|

i

?

Cancel

Reset

Save

Use the **Security** tab to assign attributes and define security labels. Click the **Add** icon (+) to add each security label selection:

- Specify a security label provider in the **Provider** field. The named provider must be loaded and must consent to the proposed labeling operation.
- Specify a security label in the **Security Label** field. The meaning of a given label is at the discretion of the label provider. PostgreSQL places no restrictions on whether or how a label provider must interpret security labels; it merely provides a mechanism for storing them.

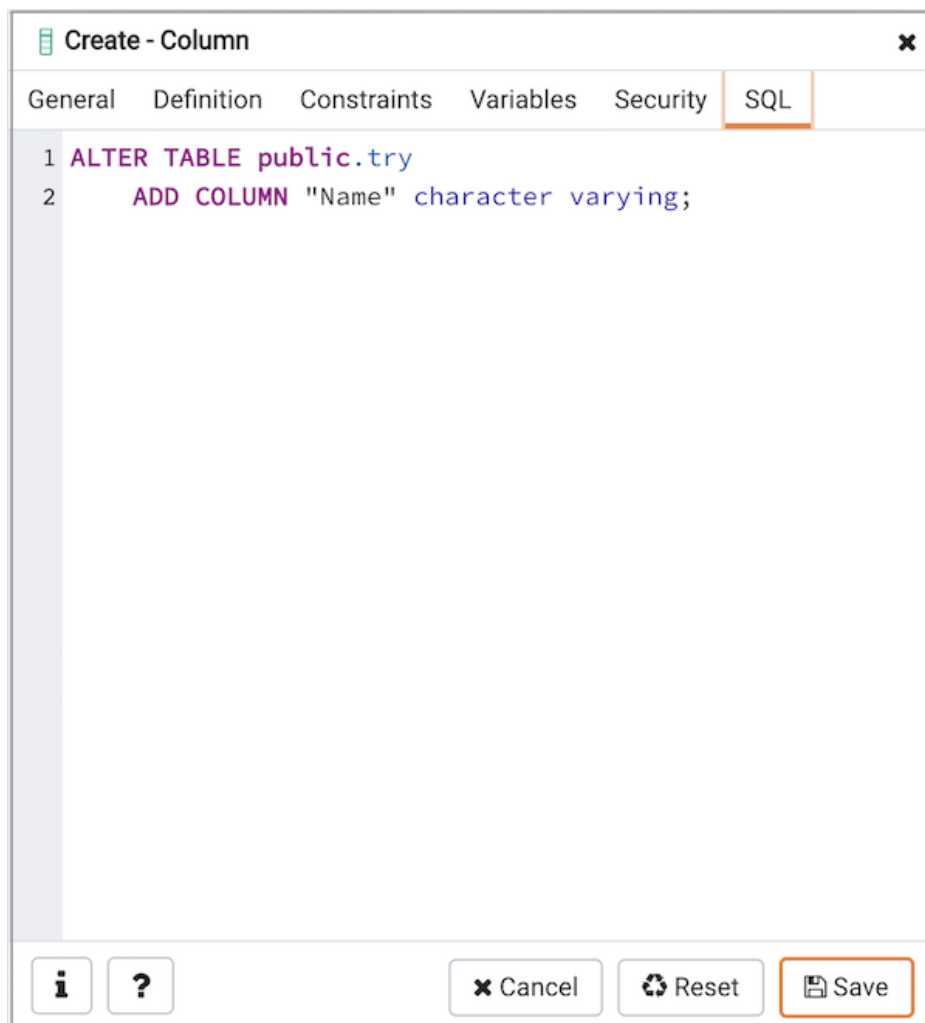
Click the **Add** icon (+) to assign additional security labels; to discard a security label, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Click the **SQL** tab to continue.

Your entries in the **Column** dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

Example

The following is an example of the sql command generated by user selections in the **Column** dialog:



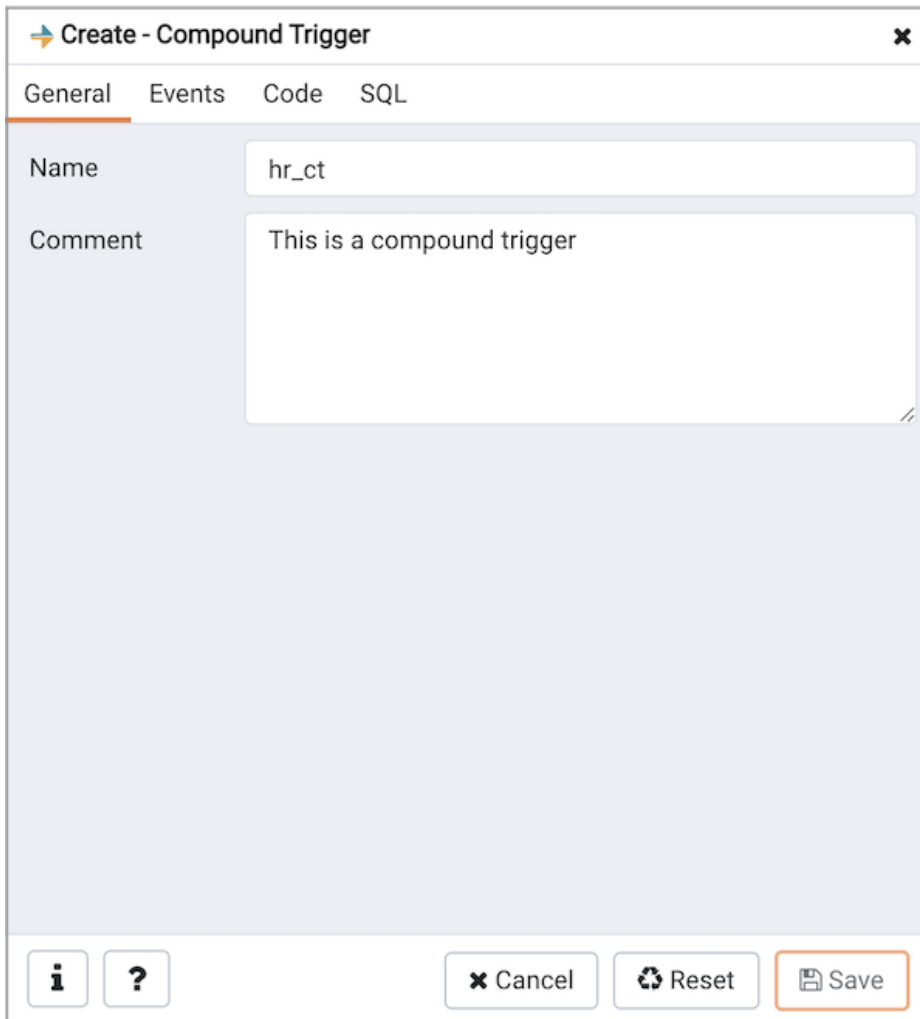
The example shown demonstrates creating a column named `territory` in the table named `distributors`.

- Click the **Info** button (i) to access online help. View context-sensitive help in the **Tabbed browser**, where a new tab displays the PostgreSQL core documentation.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

### 34.5.12.3 Compound Trigger Dialog

Use the **Compound Trigger** dialog to create a compound trigger or modify an existing compound trigger. **Compound Trigger** is supported only for EPAS server 12 and above. A compound trigger executes a specified code when certain events occur.

The **Compound Trigger** dialog organizes the development of a compound trigger through the following dialog tabs: **General**, **Events**, and **Code**. The **SQL** tab displays the SQL code generated by dialog selections.



**Create - Compound Trigger**

General Events Code SQL

Name: hr\_ct

Comment: This is a compound trigger

Buttons: [i] [?] [x] Cancel [Reset] [Save]

Use the fields in the **General** tab to identify the compound trigger:

- Use the **Name** field to add a descriptive name for the compound trigger. This must be distinct from the name of any other compound trigger for the same table. The name will be displayed in the **Browser** tree control.
- Store notes about the compound trigger in the **Comment** field.



hr\_ct

General

Events

Code

SQL

Name

hr\_ct

Trigger enabled?

Enable

Enable

Enable Replica

Enable Always

Disable

Comment

i

?

Cancel

Reset

Save

- **Trigger enabled** field is available in compound trigger dialog once the trigger is created by selecting the **properties** of the trigger. You can select one of the four options available.

Click the **Events** tab to continue.

Create - Compound Trigger

General

Events

Code

SQL

FOR Events

INSERT

Yes

UPDATE

Yes

DELETE

No

TRUNCATE

No

When

1 NEW.id > 100

Columns

x id

x name

i

?

x Cancel

Reset

Save

Use the fields in the **Events** tab to specify how and when the compound trigger fires:

- Select the type of event(s) that will invoke the compound trigger; to select an event type, move the switch next to the event to the **YES** position. The supported event types are **INSERT**, **UPDATE**, **DELETE** and **TRUNCATE**. Views cannot have TRUNCATE triggers.
- Use the **When** field to provide a boolean condition that will invoke the compound trigger.
- If defining a column-specific compound trigger, use the **Columns** field to specify the columns or columns that are the target of the compound trigger.

Click the **Code** tab to continue.

**Create - Compound Trigger**

General Events **Code** SQL

```

1 var_sal NUMBER := 10000;
2
3 BEFORE STATEMENT IS
4 BEGIN
5     var_sal := var_sal + 1000;
6     DBMS_OUTPUT.PUT_LINE('Before Statement: ' || var_sal);
7 END;
8
9 BEFORE EACH ROW IS
10 BEGIN
11     var_sal := var_sal + 1000;
12     DBMS_OUTPUT.PUT_LINE('Before Each Row: ' || var_sal);
13 END BEFORE EACH ROW;
14
15 AFTER EACH ROW IS
16 BEGIN
17     var_sal := var_sal + 1000;
18     DBMS_OUTPUT.PUT_LINE('After Each Row: ' || var_sal);
19 END AFTER EACH ROW;
20
21 AFTER STATEMENT IS
22 BEGIN
23     var_sal := var_sal + 1000;
24     DBMS_OUTPUT.PUT_LINE('After Statement: ' || var_sal);
25 END AFTER STATEMENT;
26

```

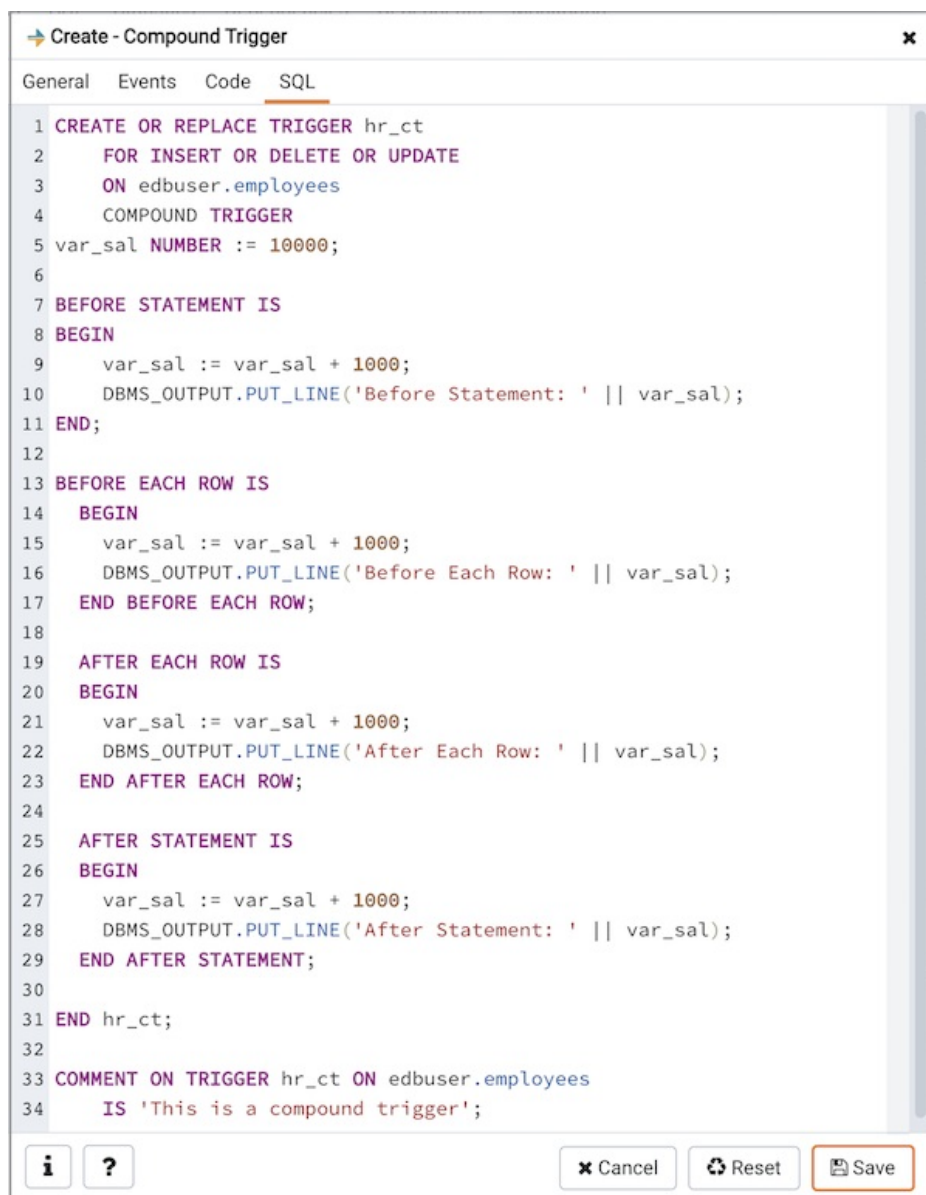
Use the **Code** field to specify the code for the five timing events **BEFORE STATEMENT**, **AFTER STATEMENT**, **BEFORE EACH ROW**, **AFTER EACH ROW**, **INSTEAD OF EACH ROW** that will be invoked when the compound trigger fires. Basic template is provided with place holders.

Click the **SQL** tab to continue.

Your entries in the **Compound Trigger** dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

## Example

The following is an example of the sql command generated by user selections in the **Compound Trigger** dialog:



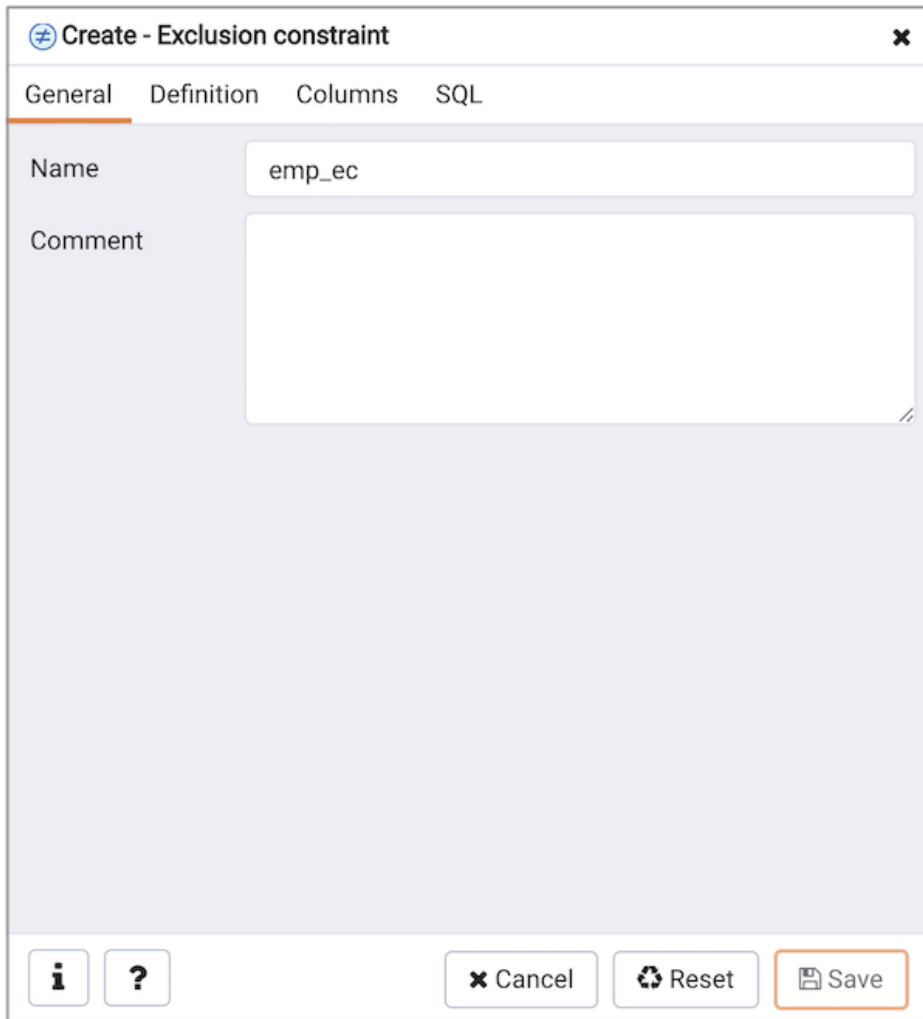
The example demonstrates creating a compound trigger named `test_ct`.

- Click the **Info** button (i) to access online help.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

### 34.5.12.4 Exclusion constraint Dialog

Use the **Exclusion constraint** dialog to define or modify the behavior of an exclusion constraint. An exclusion constraint guarantees that if any two rows are compared on the specified column or expression (using the specified operator), at least one of the operator comparisons will return false or null.

The **Exclusion constraint** dialog organizes the development of an exclusion constraint through the following dialog tabs: **General**, **Definition**, and **Columns**. The **SQL** tab displays the SQL code generated by dialog selections.



The image shows a 'Create - Exclusion constraint' dialog box with a title bar containing a plus icon and a close button. It has four tabs: 'General' (selected), 'Definition', 'Columns', and 'SQL'. The 'General' tab contains a 'Name' field with the text 'emp\_ec' and a larger 'Comment' text area. At the bottom, there are four buttons: an information icon, a question mark icon, a 'Cancel' button with a close icon, a 'Reset' button with a refresh icon, and a 'Save' button with a disk icon.

Use the fields in the **General** tab to identify the exclusion constraint:

- Use the **Name** field to provide a descriptive name for the exclusion constraint. The name will be displayed in the **Browser** tree control.

Click the **Definition** tab to continue.

**Create - Exclusion constraint**

General **Definition** Columns SQL

Tablespace: pg\_default

Access method: btree

Fill factor:

Deferrable? ☐ No

Deferred? ☐ No

Constraint:

**Buttons:** [i] [?] [x Cancel] [Reset] [Save]

Use the fields in the **Definition** tab to define the exclusion constraint:

- Use the drop-down listbox next to **Tablespace** to select the tablespace in which the index associated with the exclude constraint will reside.
- Use the drop-down listbox next to **Access method** to specify the type of index that will be used when implementing the exclusion constraint:
  - Select **gist** to specify a GiST index.
  - Select **spgist** to specify a space-partitioned GiST index.
  - Select **btree** to specify a B-tree index.
  - Select **hash** to specify a hash index.
- Use the **Fill Factor** field to specify a fill factor for the table and associated index. The fill factor is a percentage between 10 and 100. 100 (complete packing) is the default.
- Move the **Deferrable?** switch to the **Yes** position to specify that the timing of the constraint is deferrable, and can be postponed until the end of the statement. The default is **No**.
- If enabled, move the **Deferred?** switch to the **Yes** position to specify the timing of the constraint is deferred to the end of the statement. The default is **No**.
- Use the **Constraint** field to provide a condition that a row must satisfy to be included in the table.

Click the **Columns** tab to continue.

Create - Exclusion constraint

General

Definition

Columns

SQL

Columns/Expressions

+

Is expression ?

No

Column

id

Expression

|                 | Col/Exp                         | Operator class | DESC | NULLs order | Operator |
|-----------------|---------------------------------|----------------|------|-------------|----------|
| Include columns | <div>Select the column(s)</div> |                |      |             |          |

i

?

Cancel

Reset

Save

Use the fields in the *Columns* tab to specify the column(s) or expression(s) to which the constraint applies. Use the *Is expression ?* switch to enable expression text input. Use the drop-down listbox next to *Column* to select a column. Once the *Column* is selected or the *Expression* is entered then click the *Add* icon (+) to provide details of the action on the column/expression:

- The *Col/Exp* field is populated with the selection made in the *Column* drop-down listbox or the *Expression* entered.
- If applicable, use the drop-down listbox in the *Operator class* to specify the operator class that will be used by the index for the column.
- Move the *DESC* switch to *DESC* to specify a descending sort order. The default is *ASC* which specifies an ascending sort order.
- Use the *NULLs order* column to specify the placement of NULL values (when sorted). Specify *FIRST* or *LAST*.
- Use the drop-down list next to *Operator* to specify a comparison or conditional operator.

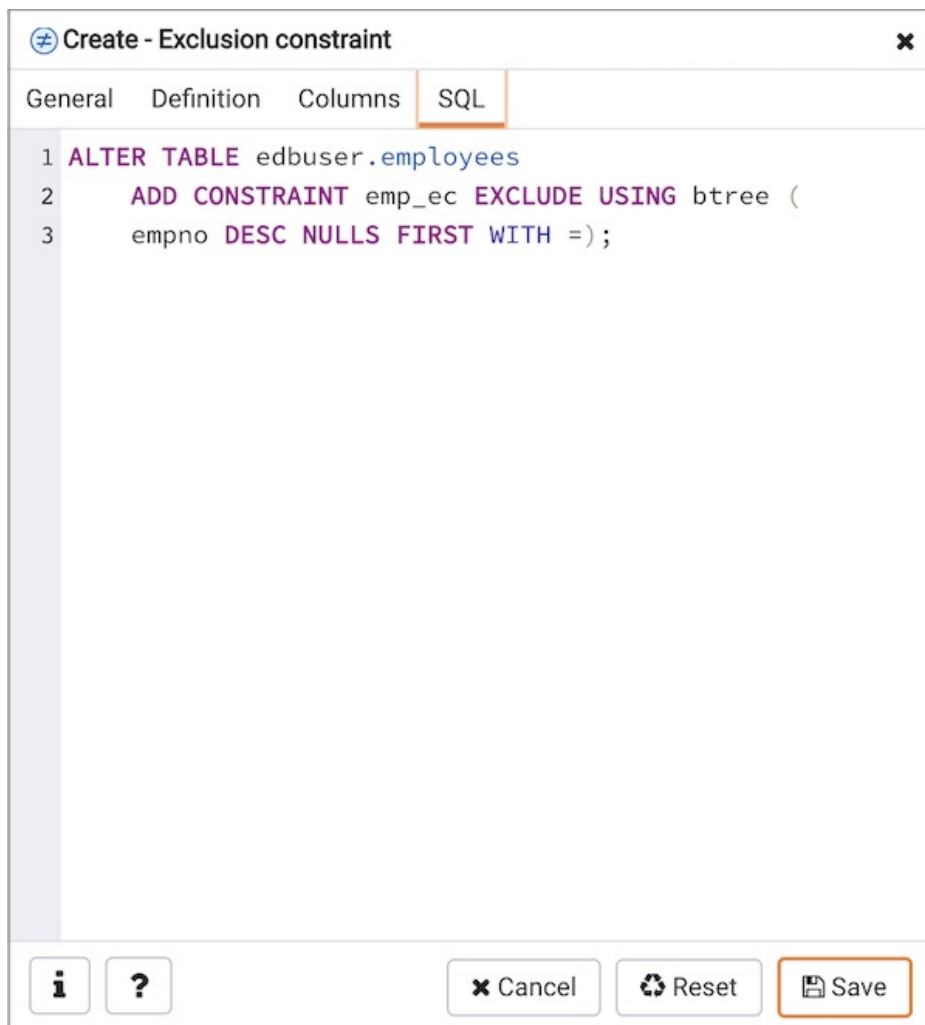
Use *Include columns* field to specify columns for *INCLUDE* clause of the constraint. This option is available in Postgres 11 and later.

Click the *SQL* tab to continue.

Your entries in the *Exclusion Constraint* dialog generate a SQL command (see an example below). Use the *SQL* tab for review; revisit or switch tabs to make any changes to the SQL command.

Example

The following is an example of the sql command generated by user selections in the *Exclusion Constraint* dialog:



The example shown demonstrates creating an exclusion constraint named `exclude_department` that restricts additions to the `dept` table to those additions that are not equal to the value of the `deptno` column. The constraint uses a btree index.



- Click the **Info** button (i) to access online help. View context-sensitive help in the **Tabbed browser**, where a new tab displays the PostgreSQL core documentation.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

### 34.5.12.5 Foreign key Dialog

Use the **Foreign key** dialog to specify the behavior of a foreign key constraint. A foreign key constraint maintains referential integrity between two tables. A foreign key constraint cannot be defined between a temporary table and a permanent table.

The **Foreign key** dialog organizes the development of a foreign key constraint through the following dialog tabs: **General**, **Definition**, **Columns**, and **Action**. The **SQL** tab displays the SQL code generated by dialog selections.



 Create - Foreign key 

General

Definition

Columns



Action


SQL


Name


emp\_deptno\_fk

Comment

 Cancel

 Reset

 Save

Use the fields in the **General** tab to identify the foreign key constraint:

- Use the **Name** field to add a descriptive name for the foreign key. The name will be displayed in the **Browser** tree control.
- Store notes about the foreign key constraint in the **Comment** field.

Click the **Definition** tab to continue.

**Create - Foreign key**

General **Definition** Columns Action SQL

Deferrable? ☒ Yes

Deferred? ☒ Yes

Match type ☒ FULL

Validated? ☒ Yes

Auto FK index? ☒ Yes

Covering index

**i ?** **Cancel** **Reset** **Save**

Use the fields in the **Definition** tab to define the foreign key constraint:

- Move the **Deferrable?** switch to the **Yes** position to specify the timing of the constraint is deferrable and can be postponed until the end of the statement. The default is **No**.
- If enabled, move the **Deferred?** switch to the **Yes** position to specify the timing of the constraint is deferred to the end of the statement. The default is **No**.
- Move the **Match type** switch specify the type of matching that is enforced by the constraint:
  - Select **Full** to indicate that all columns of a multicolumn foreign key must be null if any column is null; if all columns are null, the row is not required to have a match in the referenced table.
  - Select **Simple** to specify that a single foreign key column may be null; if any column is null, the row is not required to have a match in the referenced table.
- Move the **Validated** switch to the **Yes** position to instruct the server to validate the existing table content (against a foreign key or check constraint) when you save modifications to this dialog.
- Move the **Auto FK Index** switch to the **No** position to disable the automatic index feature.
- The field next to **Covering Index** generates the name of an index if the **Auto FK Index** switch is in the **Yes** position; or, this field is disabled.

Click the **Columns** tab to continue.

Create - Foreign key

General

Definition

Columns

Action

SQL

Columns

Local column

deptno

References

edbuser.depart...

Referencing

deptno

|  | Local  | Referenced |
|--|--------|------------|
|  | deptno | deptno     |

i

?

Cancel

Reset

Save

Use the fields in the **Columns** tab to specify one or more reference column(s). A Foreign Key constraint requires that one or more columns of a table must only contain values that match values in the referenced column(s) of a row of a referenced table:

- Use the drop-down listbox next to **Local column** to specify the column in the current table that will be compared to the foreign table.
- Use the drop-down listbox next to **References** to specify the name of the table in which the comparison column(s) resides.
- Use the drop-down listbox next to **Referencing** to specify a column in the foreign table.

Click the **Add** icon (+) to add a column to the list; repeat the steps above and click the **Add** icon (+) to add additional columns. To discard an entry, click the trash icon to the left of the entry and confirm deletion in the **Delete Row** popup.

Click the **Action** tab to continue.

Create - Foreign key

General

Definition

Columns

Action

SQL

On update

NO ACTION

On delete

NO ACTION

i

?

Cancel

Reset

Save

Use the drop-down listboxes on the **Action** tab to specify behavior related to the foreign key constraint that will be performed when data within the table is updated or deleted:

- Use the drop-down listbox next to **On update** to select an action that will be performed when data in the table is updated.
- Use the drop-down listbox next to **On delete** to select an action that will be performed when data in the table is deleted.

The supported actions are:

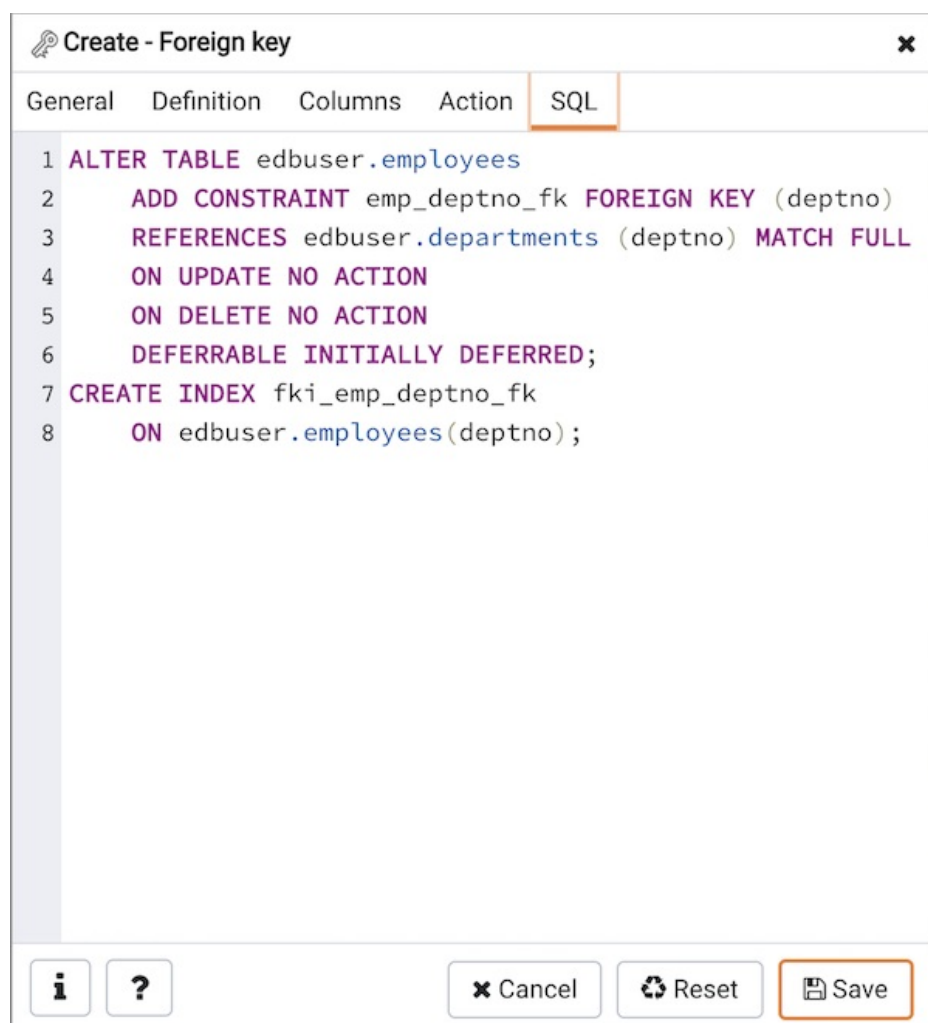
|             |   |
|-------------|---|
| NO ACTION   | Produce an error indicating that the deletion or update will create a foreign key constraint violation. If the constraint is deferred, this error will be produced at constraint check time if any referencing rows still exist. This is the default. |
| RESTRICT    | Throw an error indicating that the deletion or update would create a foreign key constraint violation. This is the same as NO ACTION except that the check is not deferrable.   |
| CASCADE     | Delete any rows referencing the deleted row, or update the values of the referencing column(s) to the new values of the referenced columns, respectively.   |
| SET NULL    | Set the referencing column(s) to null.  |
| SET DEFAULT | Set the referencing column(s) to their default values. There must be a row in the referenced table that matches the default values (if they are not null), or the operation will fail.  |

Click the **SQL** tab to continue.

Your entries in the **Foreign key** dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

Example

The following is an example of the sql command generated by user selections in the **Foreign key** dialog:



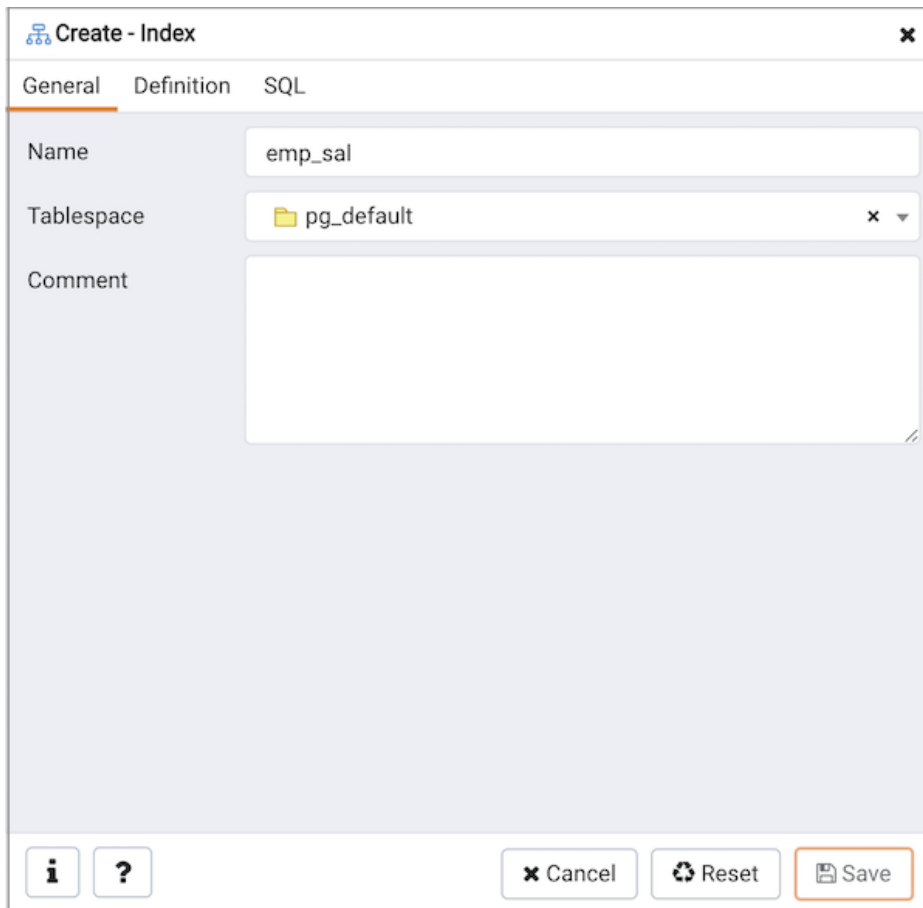
The example shown demonstrates creating a foreign key constraint named **territory\_fkey** that matches values in the **distributors** table **territory** column with those of the **sales\_territories** table **region** column.

- Click the **Info** button (i) to access online help. View context-sensitive help in the **Tabbed browser**, where a new tab displays the PostgreSQL core documentation.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

### 34.5.12.6 Index Dialog

Use the **Index** dialog to create an index on a specified table or materialized view.

The **Index** dialog organizes the development of a index through the following dialog tabs: **General** and **Definition**. The **SQL** tab displays the SQL code generated by dialog selections.



The image shows a 'Create - Index' dialog box with three tabs: 'General', 'Definition', and 'SQL'. The 'General' tab is selected. It contains three fields: 'Name' with the value 'emp\_sal', 'Tablespace' with a dropdown menu showing 'pg\_default', and a large 'Comment' text area. At the bottom, there are buttons for 'Cancel', 'Reset', and 'Save' (which is highlighted with an orange border). There are also information and help icons on the bottom left.

Use the fields in the **General** tab to identify the index:

- Use the **Name** field to add a descriptive name for the index. The name will be displayed in the **Browser** tree control.
- Use the drop-down listbox next to **Tablespace** to select the tablespace in which the index will reside.
- Store notes about the index in the **Comment** field.

Click the **Definition** tab to continue.

**Create - Index**

General **Definition** SQL

Access Method: btree

Fill factor:

Unique? ☒ Yes

Clustered? ☒ Yes

Concurrent build? ☐ No

Constraint: 1

Columns

|  | Column | Operator class    | Sort order | NULLs | Collation         |
|--|--------|-------------------|------------|-------|-------------------|
|  | sal    | Select an item... | ASC        | LAST  | Select an item... |

Include columns: Select the column(s)

Buttons: [i] [?] [Cancel] [Reset] [Save]

Use the fields in the **Definition** tab to define the index:

- Use the drop-down listbox next to **Access Method** to select an index type:
  - Select **btree** to create a B-tree index. A B-tree index may improve performance when managing equality and range queries on data that can be sorted into some ordering (the default).
  - Select **hash** to create a hash index. A hash index may improve performance when managing simple equality comparisons.
  - Select **gist** to create a GiST index. A GiST index may improve performance when managing two-dimensional geometric data types and nearest-neighbor searches
  - Select **gin** to create a GIN index. A GIN index may performance when managing values with more than one key.
  - Select **spgist** to create a space-partitioned GiST index. A SP-GiST index may improve performance when managing non-balanced data structures.
  - Select **brin** to create a BRIN index. A BRIN index may improve performance when managing minimum and maximum values and ranges.
- Use the **Fill Factor** field to specify a fill factor for the index. The fill factor specifies how full the selected method will try to fill each index page.
- Move the **Unique?** switch to the **Yes** position to check for duplicate values in the table when the index is created and when data is added. The default is **No**.
- Move the **Clustered?** switch to the **Yes** position to instruct the server to cluster the table.
- Move the **Concurrent build?** switch to the **Yes** position to build the index without taking any locks that prevent concurrent inserts, updates, or deletes on the table.
- Use the **Constraint** field to provide a constraint expression; a constraint expression limits the entries in the index to those rows that satisfy the constraint.

Use the context-sensitive fields in the **Columns** panel to specify which column(s) the index queries. Click the **Add** icon (+) to add a column:

- Use the drop-down listbox in **Column** field to select the name of the column from the table.
- If enabled, use the drop-down listbox to select an available **Operator class** to specify the type of action performed on the column.
- If enabled, move the **Sort order** switch to specify the sort order:
  - Select **ASC** to specify an ascending sort order (the default);
  - Select **DESC** to specify a descending sort order.
- If enabled, move the **Nulls** switch to specify the sort order of nulls:
  - Select **First** to specify nulls sort before non-nulls;
  - Select **Last** to specify nulls sort after non-nulls (the default).
- Use the drop-down listbox in the **Collation** field to select a collation to use for the index.

Use **Include columns** field to specify columns for **INCLUDE** clause of the index. This option is available in Postgres 11 and later.

Click the **SQL** tab to continue.

Your entries in the **Index** dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

### Example

The following is an example of the sql command generated by user selections in the **Index** dialog:

```

1 CREATE UNIQUE INDEX emp_sal
2   ON edbuser.employees USING btree
3   (sal ASC NULLS LAST)
4   TABLESPACE pg_default;
5
6 ALTER TABLE edbuser.employees
7   CLUSTER ON emp_sal;
  
```

The example shown demonstrates creating an index named **dist\_codes** that indexes the values in the **code** column of the **distributors** table.



- Click the **Info** button (i) to access online help. View context-sensitive help in the **Tabbed browser**, where a new tab displays the PostgreSQL core documentation.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

### 34.5.12.7 Primary key Dialog

Use the **Primary key** dialog to create or modify a primary key constraint. A primary key constraint indicates that a column, or group of columns, uniquely identifies rows in a table. This requires that the values in the selected column(s) be both unique and not null.

The **Primary key** dialog organizes the development of a primary key constraint through the **General** and **Definition** tabs. The **SQL** tab displays the SQL code generated by dialog selections.

**Create - Primary key** [X]

General Definition SQL

Name: emp\_empno\_pk

Comment: [Text Area]

[i] [?] [Cancel] [Reset] [Save]

Use the fields in the **General** tab to identify the primary key:

- Use the **Name** field to add a descriptive name for the primary key constraint. The name will be displayed in the **Browser** tree control.

Click the **Definition** tab to continue.

Use the fields in the **Definition** tab to define the primary key constraint:

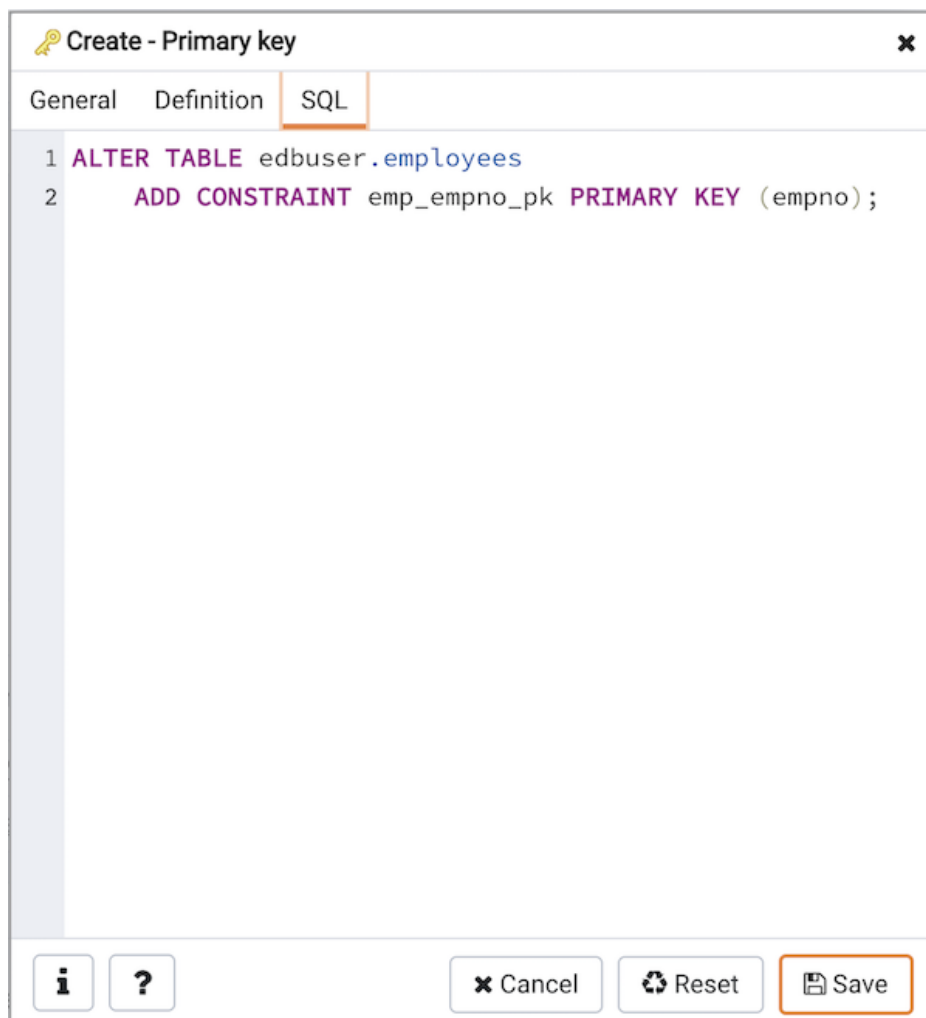
- Click inside the **Columns** field and select one or more column names from the drop-down listbox. To delete a selection, click the **x** to the left of the column name. The primary key constraint should be different from any unique constraint defined for the same table; the selected column(s) for the constraints must be distinct.
- Use **Include columns** field to specify columns for **INCLUDE** clause of the index. This option is available in Postgres 11 and later.
- Select the name of the tablespace in which the primary key constraint will reside from the drop-down listbox in the **Tablespace** field.
- Select the name of an index from the drop-down listbox in the **Index** field. This field is optional. Adding a primary key will automatically create a unique B-tree index on the column or group of columns listed in the primary key, and will force the column(s) to be marked NOT NULL.
- Use the **Fill Factor** field to specify a fill factor for the table and index. The fill factor for a table is a percentage between 10 and 100. 100 (complete packing) is the default.
- Move the **Deferrable?** switch to the **Yes** position to specify the timing of the constraint is deferrable and can be postponed until the end of the statement. The default is **No**.
- If enabled, move the **Deferred?** switch to the **Yes** position to specify the timing of the constraint is deferred to the end of the statement. The default is **No**.

Click the **SQL** tab to continue.

Your entries in the **Primary key** dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

#### Example

The following is an example of the sql command generated by user selections in the **Primary key** dialog:



The example shown demonstrates creating a primary key constraint named `dept_pkey` on the `dept_id` column of the `dept` table.

- Click the **Info** button (i) to access online help. View context-sensitive help in the **Tabbed browser**, where a new tab displays the PostgreSQL core documentation.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

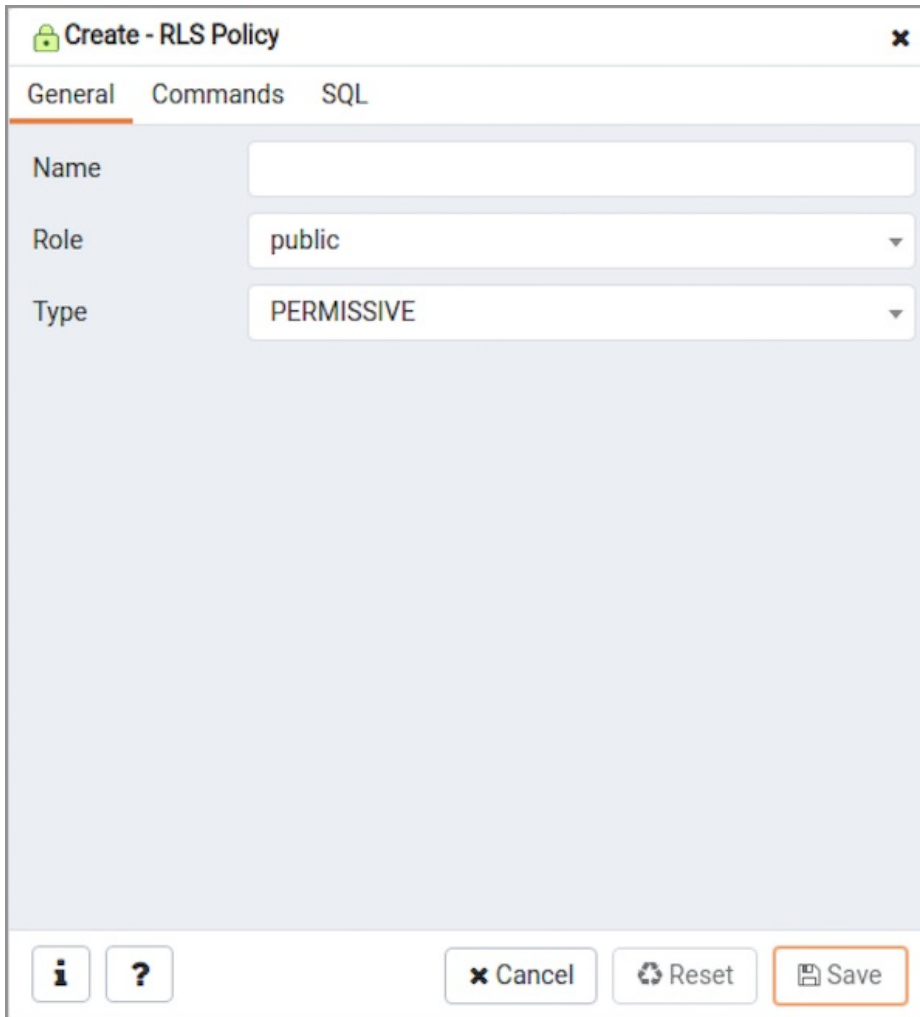
### 34.5.12.8 RLS Policy Dialog

Use the **RLS Policy** dialog to Create a Row Level Security Policy.

#### Note

If the Row Level Security is enabled at table level and no policy is created then by default **Deny Policy** is applied. That means, no rows are visible or can be modified for that table.

The **RLS Policy** dialog creates a Row Level Security Policy through the following dialog tabs: **General**, and **Commands**. The **SQL** tab displays the SQL code generated by dialog selections.



The image shows a 'Create - RLS Policy' dialog box with three tabs: 'General', 'Commands', and 'SQL'. The 'General' tab is selected. It contains three fields: 'Name' (a text input field), 'Role' (a dropdown menu with 'public' selected), and 'Type' (a dropdown menu with 'PERMISSIVE' selected). At the bottom, there are four buttons: an information icon, a help icon, a 'Cancel' button, a 'Reset' button, and a 'Save' button (which is highlighted with an orange border).

Use the fields in the **General** tab to define the RLS Policy:

- Use the **Name** field to add a descriptive name for the RLS Policy. The name will be displayed in the Browser tree control.
- Use the drop-down listbox next to **Role** to select the Role to which the RLS Policy is to be applied.
- Use the drop-down listbox next to **Type** to select the type of the policy.

Click the **Commands** tab to continue.

**Create - RLS Policy**

General **Commands** SQL

Event: ALL

Using: 1

With check: 1

**Using:** This expression will be added to queries that refer to the table if row level security is enabled. Rows for which the expression returns true will be visible. Any rows for which the expression returns false or null will not be visible to the user (in a SELECT), and will not be available for modification (in an UPDATE or DELETE). Such rows are silently suppressed; no error is reported.

**With check:** This expression will be used in INSERT and UPDATE queries against the table if row level security is enabled. Only rows for which the expression evaluates to true will be allowed. An error will be thrown if the expression evaluates to false or null for any of the records inserted or any of the records that result from the update.

Buttons: i, ?, x Cancel, Reset, Save

Use the fields in the **Commands** tab to define the RLS Policy:

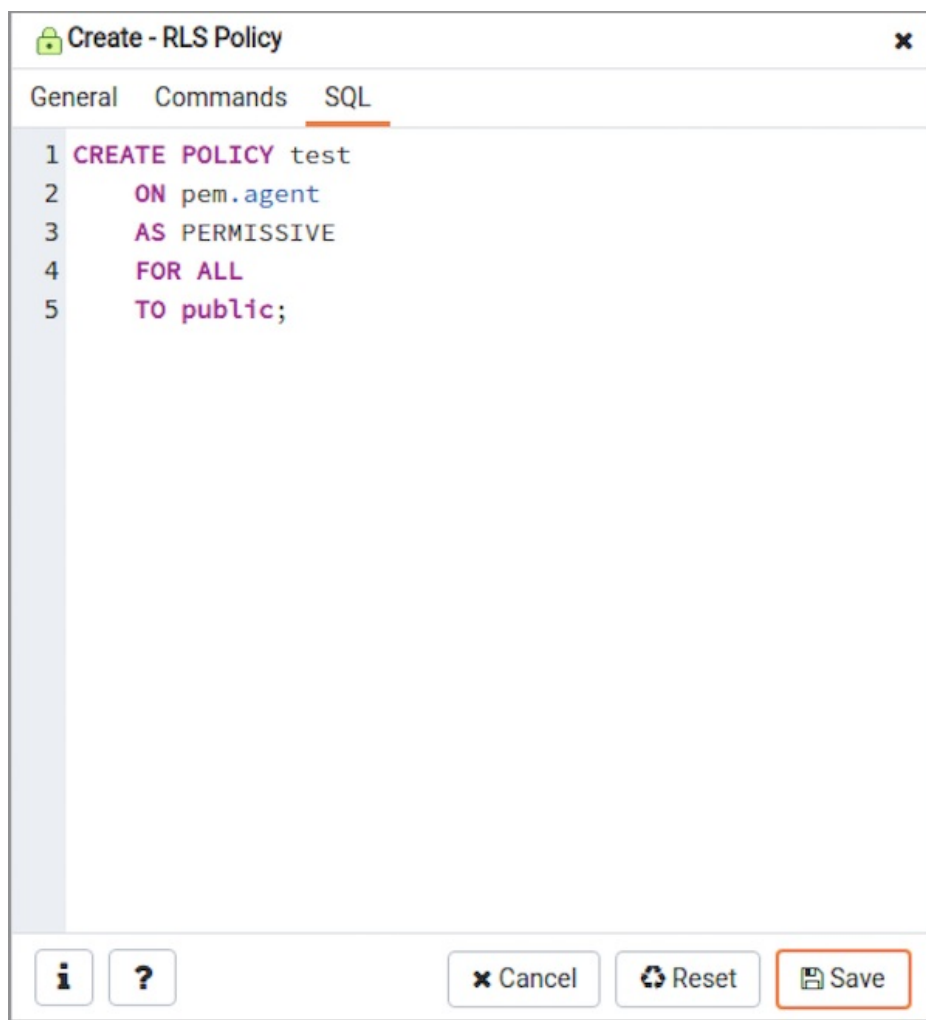
- Use the drop-down listbox next to **Event** to select the command to which policy applies. Valid options are ALL, SELECT, INSERT, UPDATE, and DELETE. Default is ALL.
- Use the **Using** field to add a SQL conditional expression returning boolean. This expression will be added to queries that refer to the table if row level security is enabled.
- Use the **With check** field to add a SQL conditional expression returning boolean. This expression will be used in INSERT and UPDATE queries against the table if row level security is enabled.

Click the **SQL** tab to continue.

Your entries in the **RLS Policy** dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

## Example

The following is an example of the sql command generated by user selections in the **RLS Policy** dialog:



The example shown demonstrates creating a RLS Policy named `account_managers` that applies the Row Level Security on the `accounts` table.

- Click the **Info** button (i) to access online help.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

### 34.5.12.9 Rule Dialog

Use the **Rule** dialog to define or modify a rule for a specified table or view. A PostgreSQL rule allows you to define an additional action that will be performed when a SELECT, INSERT, UPDATE, or DELETE is performed against a table.

The **Rule** dialog organizes the development of a rule through the **General**, **Definition**, **Condition**, **Commands** tabs. The **SQL** tab displays the SQL code generated by dialog selections.

Create - rule

General

Definition

Condition

Commands

SQL

Name

notify\_me

Comment

i

?

Cancel

Reset

Save

Use the fields in the **General** tab to identify the rule:

- Use the **Name** field to add a descriptive name for the rule. The name will be displayed in the **Browser** tree control. Multiple rules on the same table are applied in alphabetical name order.
- Store notes about the rule in the **Comment** field.

Click the **Definition** tab to continue.

**Create - rule**

General **Definition** Condition Commands SQL

Event UPDATE

Do instead? Yes

**i** **?** ✕ Cancel 🔄 Reset 💾 Save

Use the fields in the **Definition** tab to write parameters:

- Click inside the **Event** field to select the type of event that will invoke the rule; event may be **Select**, **Insert**, **Update**, or **Delete**.
- Move the **Do Instead** switch to **Yes** indicate that the commands should be executed instead of the original command; if Do Instead specifies **No**, the rule will be invoked in addition to the original command.

Click the **Condition** tab to continue.



Create - rule

General

Definition

Condition

Commands

SQL

1

i

?

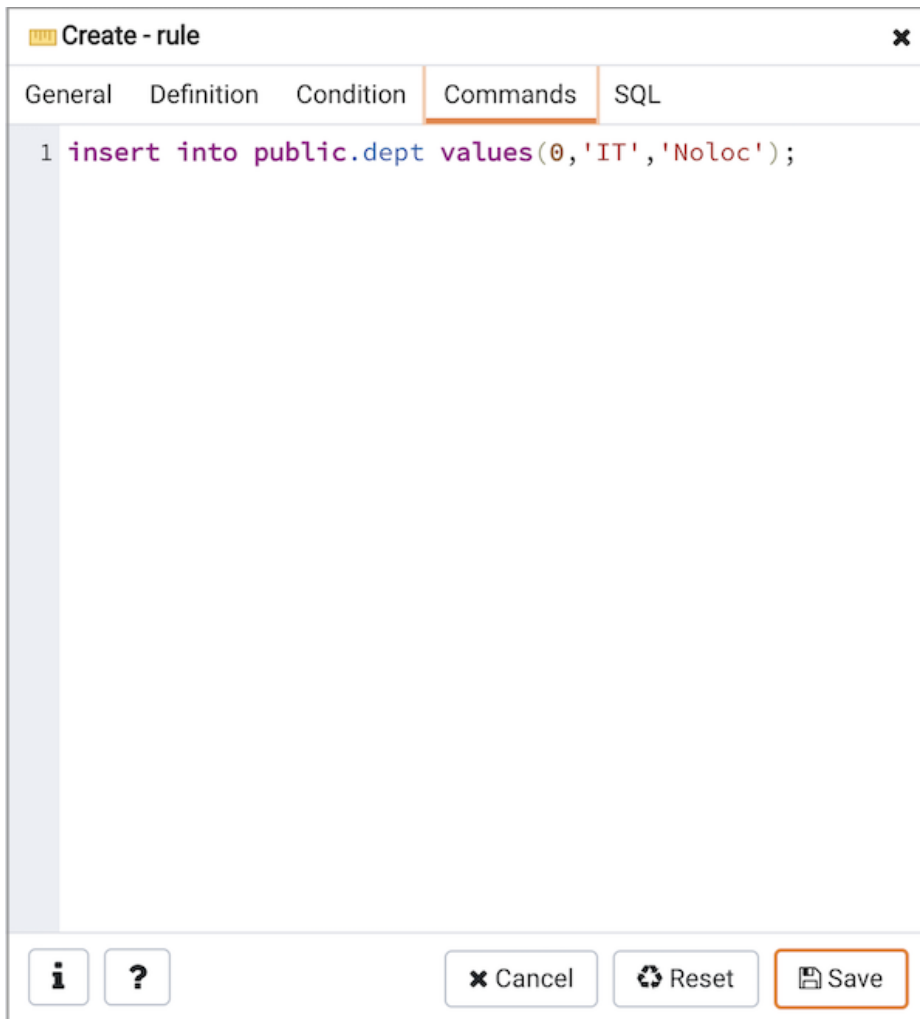
✕ Cancel

↺ Reset

💾 Save

Specify a SQL conditional expression that returns a boolean value in the editor.

Click the **Commands** tab to continue.



**Create - rule**

General Definition Condition **Commands** SQL

```
1 insert into public.dept values(0, 'IT', 'NoLoc');
```

**i** **?** **Cancel** **Reset** **Save**

Provide a command in the editor that defines the action performed by the rule.

Click the **SQL** tab to continue.

Your entries in the **Rule** dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

### Example

The following is an example of the sql command generated by user selections in the **Rule** dialog:



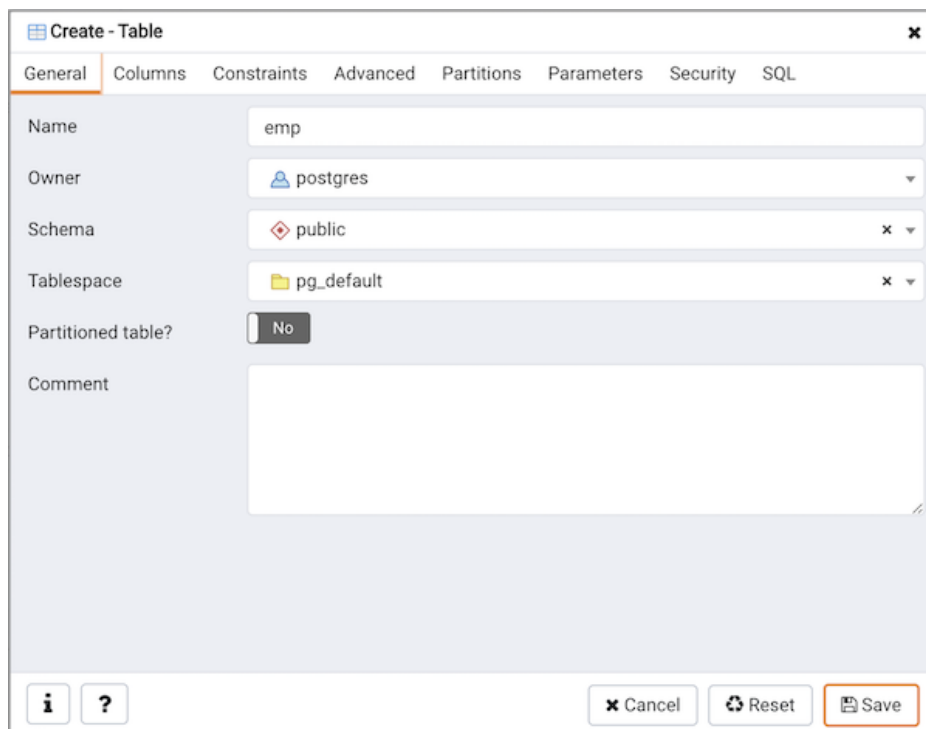
The example sends a notification when an UPDATE executes against a table.

- Click the **Info** button (i) to access online help. View context-sensitive help in the **Tabbed browser**, where a new tab displays the PostgreSQL core documentation.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

### 34.5.12.10 Table Dialog

Use the **Table** dialog to create or modify a table.

The **Table** dialog organizes the development of a table through the following dialog tabs: **General**, **Columns**, **Constraints**, **Advanced**, **Parameter**, and **Security**. The **SQL** tab displays the SQL code generated by dialog selections.



**Create - Table**

General Columns Constraints Advanced Partitions Parameters Security SQL

Name: emp

Owner: postgres

Schema: public

Tablespace: pg\_default

Partitioned table?: No

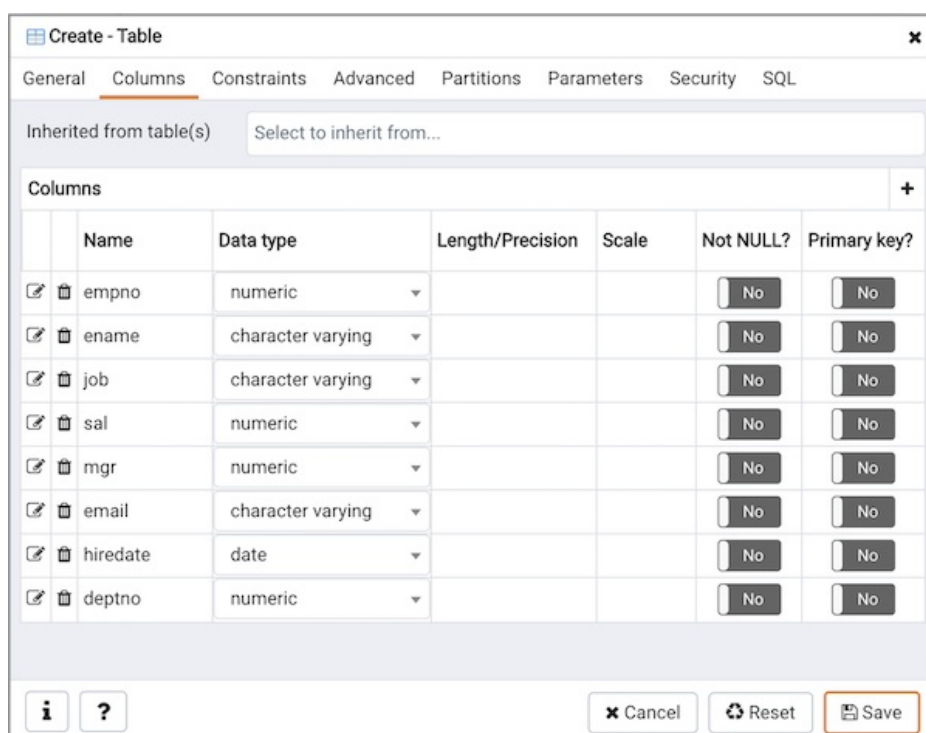
Comment:

Buttons: Cancel, Reset, Save

Use the fields in the **General** tab to identify the table:

- Use the **Name** field to add a descriptive name for the table. A table cannot have the same name as any existing table, sequence, index, view, foreign table, or data type in the same schema. The name specified will be displayed in the **Browser** tree control. This field is required.
- Select the owner of the table from the drop-down listbox in the **Owner** field. By default, the owner of the table is the role that creates the table.
- Select the name of the schema in which the table will reside from the drop-down listbox in the **Schema** field.
- Use the drop-down listbox in the **Tablespace** field to specify the tablespace in which the table will be stored.
- Move the **Partitioned Table?** switch to the **Yes** in case you want to create a partitioned table. Option is available for PostgreSQL 10 and above.
- Store notes about the table in the **Comment** field.

Click the **Columns** tab to continue.



**Create - Table**

General Columns Constraints Advanced Partitions Parameters Security SQL

Inherited from table(s): Select to inherit from...

Columns

|  | Name     | Data type         | Length/Precision | Scale | Not NULL? | Primary key? |
|--|----------|-------------------|------------------|-------|-----------|--------------|
|  | empno    | numeric           |                  |       | No        | No           |
|  | ename    | character varying |                  |       | No        | No           |
|  | job      | character varying |                  |       | No        | No           |
|  | sal      | numeric           |                  |       | No        | No           |
|  | mgr      | numeric           |                  |       | No        | No           |
|  | email    | character varying |                  |       | No        | No           |
|  | hiredate | date              |                  |       | No        | No           |
|  | deptno   | numeric           |                  |       | No        | No           |

Buttons: Cancel, Reset, Save

Use the drop-down listbox next to **Inherited from table(s)** to specify any parent table(s); the table will inherit columns from the selected parent table(s). Click inside the **Inherited from table(s)** field to select a table name from a drop-down list. Repeat to add any other parent tables. Delete a selected table by clicking the **x** to the left of the parent name. Note that inherited column names and datatypes are not editable in the current dialog; they must be modified at the parent level.

Click the **Add** icon (+) to specify the names of columns and their datatypes in the **Columns** table:

- Use the **Name** field to add a descriptive name for the column.
- Use the drop-down listbox in the **Data type** field to select a data type for the column. This can include array specifiers. For more information on the data types supported by PostgreSQL, refer to Chapter 8 of the core documentation.
- If enabled, use the **Length** and **Precision** fields to specify the maximum number of significant digits in a numeric value, or the maximum number of characters in a text value.
- Move the **Not NULL?** switch to the **Yes** position to require a value in the column field.
- Move the **Primary key?** switch to the **Yes** position to specify the column is the primary key constraint.

Click the **Add** icon (+) to add additional columns; to discard a column, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Click the **Constraints** tab to continue.

Create - Table

General

Columns

Constraints

Advanced

Partitions

Parameters

Security

SQL



Primary Key

Foreign Key

Check

Unique

Exclude

|   | Name         | Columns |
|---|--------------|---------|
|   | emp_empno_pk | empno   |

General

Definition

Name

emp\_empno\_pk

Comment

?

?

Cancel

Reset

Save

Use the fields in the **Constraints** tab to provide a table or column constraint. Optional constraint clauses specify constraints (tests) that new or updated rows must satisfy for an **INSERT** or **UPDATE** operation to succeed. Select the appropriate constraint type by selecting one of the following tabs on the **Constraints** panel:

| Tab Name    | Constraint   |
|-------------|--|
| Primary Key | Provides a unique identifier for each row in the table.  |
| Foreign Key | Maintains referential integrity between two tables.  |
| Check       | Requires data satisfies an expression or condition before insertion or modification.                           |
| Unique      | Ensures that the data contained in a column, or a group of columns, is unique among all the rows in the table. |



Create - Table

General

Columns

Constraints

Advanced

Partitions

Parameters

Security

SQL

Primary Key

Foreign Key

Check

Unique

Exclude

emp\_deptno\_fk

Name

Columns

General

Definition

Columns

Action

Name

emp\_deptno\_fk

Comment

i

?

Cancel

Reset

Save

To add a foreign key constraint, select the **Foreign Key** tab, and click the **Add** icon (+). To define the constraint, click the **Edit** icon to the left of the **Trash** icon. A dialog similar to the **Foreign key** dialog (accessed by right clicking on **Constraints** in the **Browser** tree control) opens.

Use the fields in the **General** tab to identify the foreign key constraint:

- Use the **Name** field to add a descriptive name for the foreign key constraint. The name will be displayed in the **Browser** tree control.
- Provide notes about the foreign key in the **Comment** field.

Click the **Definition** tab to continue.

Create - Table

General

Columns

Constraints

Advanced

Partitions

Parameters

Security

SQL

Primary Key

Foreign Key

Check

Unique

Exclude

| Name          | Columns |
|---------------|---------|
| emp_deptno_fk |         |

General

Definition

Columns

Action

Deferrable?

Yes

Deferred?

Yes

Match type

FULL

Validated?

Yes

Auto FK index?

No

Covering index

?

?

Cancel

Reset

Save

Use the fields in the **Definition** tab to define the foreign key constraint:

- Move the **Deferrable?** switch to the **Yes** position to specify the timing of the constraint is deferrable and can be postponed until the end of the statement. The default is **No**.
- If enabled, move the **Deferred?** switch to the **Yes** position to specify the timing of the constraint is deferred to the end of the statement. The default is **No**.
- Move the **Match type** switch specify the type of matching that is enforced by the constraint:
  - Select **Full** to indicate that all columns of a multicolumn foreign key must be null if any column is null; if all columns are null, the row is not required to have a match in the referenced table.
  - Select **Simple** to specify that a single foreign key column may be null; if any column is null, the row is not required to have a match in the referenced table.
- Move the **Validated** switch to the **Yes** position to instruct the server to validate the existing table content (against a foreign key or check constraint) when you save modifications to this dialog.
- Move the **Auto FK Index** switch to the **No** position to disable the automatic index feature.
- The field next to **Covering Index** generates the name of an index if the **Auto FK Index** switch is in the **Yes** position; or, this field is disabled.

Click the **Columns** tab to continue.

The screenshot shows the 'Create - Table' dialog box with the 'Columns' tab selected. The 'Foreign Key' sub-tab is active, showing a list of constraints with 'emp\_deptno\_fk' selected. Below this, the 'Columns' sub-tab is active, showing the configuration for the selected constraint. The 'Local column' is 'deptno', the 'References' table is 'public.dept', and the 'Referencing' column is 'deptno'. A table at the bottom shows the mapping between the local and referenced columns.

| Local  | Referenced |
|--------|------------|
| deptno | deptno     |

Use the fields in the **Columns** tab to specify one or more reference column(s). A Foreign Key constraint requires that one or more columns of a table must only contain values that match values in the referenced column(s) of a row of a referenced table:

- Use the drop-down listbox next to **Local column** to specify the column in the current table that will be compared to the foreign table.
- Use the drop-down listbox next to **References** to specify the name of the table in which the comparison column(s) resides.
- Use the drop-down listbox next to **Referencing** to specify a column in the foreign table.



Click the **Add** icon (+) to add a column to the list; repeat the steps above and click the **Add** icon (+) to add additional columns. To discard an entry, click the trash icon to the left of the entry and confirm deletion in the **Delete Row** popup.

Click the **Action** tab to continue.

Create - Table

General

Columns

Constraints

Advanced

Partitions

Parameters

Security

SQL

Primary Key

Foreign Key

Check

Unique

Exclude

| Name  | Columns              |
|---|----------------------|
| <div> <div></div> <div>emp_deptno_fk</div> </div> | (deptno) -> (deptno) |

General

Definition

Columns

Action

On update

NO ACTION

On delete

NO ACTION

?

Cancel

Reset

Save

Use the drop-down listboxes on the **Action** tab to specify behavior related to the foreign key constraint that will be performed when data within the table is updated or deleted:

- Use the drop-down listbox next to **On update** to select an action that will be performed when data in the table is updated.
- Use the drop-down listbox next to **On delete** to select an action that will be performed when data in the table is deleted.

The supported actions are:

|             |   |
|-------------|---|
| NO ACTION   | Produce an error indicating that the deletion or update will create a foreign key constraint violation. If the constraint is deferred, this error will be produced at constraint check time if any referencing rows still exist. This is the default. |
| RESTRICT    | Throw an error indicating that the deletion or update would create a foreign key constraint violation. This is the same as NO ACTION except that the check is not deferrable.   |
| CASCADE     | Delete any rows referencing the deleted row, or update the values of the referencing column(s) to the new values of the referenced columns, respectively.   |
| SET NULL    | Set the referencing column(s) to null.  |
| SET DEFAULT | Set the referencing column(s) to their default values. There must be a row in the referenced table that matches the default values (if they are not null), or the operation will fail.  |

The screenshot shows the 'Create - Table' dialog with the 'Constraints' tab selected. Under 'Constraints', the 'Check' sub-tab is active. A table lists the constraint 'emp\_sal\_ck'. Below this, the 'General' sub-tab is open, showing a 'Name' field with 'emp\_sal\_ck' and an empty 'Comment' text area. At the bottom are 'Cancel', 'Reset', and 'Save' buttons.

| Primary Key |  | Foreign Key |  | Check |  | Unique |  | Exclude |  |
|-------------|--|-------------|--|-------|--|--------|--|---------|--|
|             |  |             |  |       |  |        |  |         |  |
|             |  |             |  |       |  |        |  |         |  |

| General |            | Definition |  |
|---------|------------|------------|--|
| Name    | emp_sal_ck |            |  |
| Comment |            |            |  |

To add a check constraint, select the **Check** tab on the panel, and click the **Add** icon (+). To define the check constraint, click the **Edit** icon to the left of the **Trash** icon. A dialog similar to the **Check** dialog (accessed by right clicking on **Constraints** in the **Browser** tree control) opens.

Use the fields in the **General** tab to identify the check constraint:

- Use the **Name** field to add a descriptive name for the check constraint. The name will be displayed in the **Browser** tree control. With PostgreSQL 9.5 forward, when a table has multiple check constraints, they will be tested for each row in alphabetical order by name and after NOT NULL constraints.
- Provide notes about the check constraint in the **Comment** field.

Click the **Definition** tab to continue.

Create - Table

GeneralColumnsConstraintsAdvancedPartitionsParametersSecuritySQL

Primary KeyForeign KeyCheckUniqueExclude

Name

emp\_sal\_ck

Check

sal > 0

GeneralDefinition

Check

sal > 0

No inherit?

No

Don't validate?

Yes

i?

CancelResetSave

Use the fields in the **Definition** tab to define the check constraint:

- Provide the expression that a row must satisfy in the **Check** field. This field is required.
- Move the **No Inherit?** switch to the **Yes** position to specify this constraint is automatically inherited by a table's children. The default is **No**.
- Move the **Don't validate?** switch to the **No** position to skip validation of existing data; the constraint may not hold for all rows in the table. The default is **Yes**.

Create - Table

GeneralColumnsConstraintsAdvancedPartitionsParametersSecuritySQL

Primary KeyForeign KeyCheckUniqueExclude

Name

emp\_email\_uk

Columns

GeneralDefinition

Name

emp\_email\_uk

Comment

i?

CancelResetSave

To add a unique constraint, select the **Unique** tab on the panel, and click the **Add** icon (+). To define the constraint, click the **Edit** icon to the left of the **Trash** icon. A dialog similar to the **Unique constraint** dialog (accessed by right clicking on **Constraints** in the **Browser** tree control) opens.

Use the fields in the **General** tab to identify the unique constraint:

- Use the **Name** field to add a descriptive name for the unique constraint. The name will be displayed in the **Browser** tree control.
- Provide notes about the unique constraint in the **Comment** field.

Click the **Definition** tab to continue.

The screenshot shows the 'Create - Table' dialog box with the 'Constraints' tab selected. Under the 'Unique' sub-tab, a table lists the constraint details:

| Name         | Columns |
|--------------|---------|
| emp_email_uk | email   |

Below this table, the 'Definition' sub-tab is active, showing configuration options for the unique constraint:

- Columns:** A text box containing 'x email'.
- Include columns:** A text box with the placeholder 'Select the column(s)'.
- Tablespace:** A dropdown menu with the placeholder 'Select an item...'.
- Fill factor:** A text box.
- Deferrable?:** A toggle switch set to 'No'.
- Deferred?:** A toggle switch set to 'No'.

At the bottom of the dialog are buttons for 'Cancel', 'Reset', and 'Save'.

Use the fields in the **Definition** tab to define the unique constraint:

- Click inside the **Columns** field and select one or more column names from the drop-down listbox. To delete a selection, click the **x** to the left of the column name. The unique constraint should be different from the primary key constraint defined for the same table; the selected column(s) for the constraints must be distinct.
- Select the name of the tablespace in which the unique constraint will reside from the drop-down listbox in the **Tablespace** field.
- Use the **Fill Factor** field to specify a fill factor for the table and index. The fill factor for a table is a percentage between 10 and 100. 100 (complete packing) is the default.
- Move the **Deferrable?** switch to the **Yes** position to specify the timing of the constraint is deferrable and can be postponed until the end of the statement. The default is **No**.
- If enabled, move the **Deferred?** switch to the **Yes** position to specify the timing of the constraint is deferred to the end of the statement. The default is **No**.

**Create - Table**

General Columns Constraints Advanced Partitions Parameters Security SQL

Primary Key Foreign Key Check Unique Exclude

|                                     | Name       | Columns |
|-------------------------------------|------------|---------|
| <input checked="" type="checkbox"/> | emp_mgr_ek | mgr     |

General Definition Columns

Name

Comment

Cancel Reset Save

To add an exclusion constraint, select the **Exclude** tab on the panel, and click the **Add** icon (+). To define the constraint, click the **Edit** icon to the left of the **Trash** icon. A dialog similar to the **Exclusion constraint** dialog (accessed by right clicking on **Constraints** in the **Browser** tree control) opens.

Use the fields in the **General** tab to identify the exclusion constraint:

- Use the **Name** field to provide a descriptive name for the exclusion constraint. The name will be displayed in the **Browser** tree control.
- Provide notes about the exclusion constraint in the **Comment** field.

Click the **Definition** tab to continue.

The screenshot shows the 'Create - Table' dialog with the 'Constraints' tab selected. Under the 'Exclude' sub-tab, a table lists constraints, with 'emp\_job\_ek' selected. Below this, the 'Definition' sub-tab is active, showing configuration options for the exclusion constraint: Tablespace (pg\_default), Access method (btree), Fill factor (empty), Deferrable? (No), and Deferred? (No). The 'Constraint' field is empty. At the bottom are buttons for 'Cancel', 'Reset', and 'Save'.

Use the fields in the **Definition** tab to define the exclusion constraint:

- Use the drop-down listbox next to **Tablespace** to select the tablespace in which the index associated with the exclude constraint will reside.
- Use the drop-down listbox next to **Access method** to specify the type of index that will be used when implementing the exclusion constraint:
  - Select **gist** to specify a GiST index (the default).
  - Select **spgist** to specify a space-partitioned GiST index.
  - Select **btree** to specify a B-tree index.
  - Select **hash** to specify a hash index.
- Use the **Fill Factor** field to specify a fill factor for the table and associated index. The fill factor is a percentage between 10 and 100. 100 (complete packing) is the default.
- Move the **Deferrable?** switch to the **Yes** position to specify that the timing of the constraint is deferrable, and can be postponed until the end of the statement. The default is **No**.
- If enabled, move the **Deferred?** switch to the **Yes** position to specify the timing of the constraint is deferred to the end of the statement. The default is **No**.
- Use the **Constraint** field to provide a condition that a row must satisfy to be included in the table.

Click the **Columns** tab to continue.

Create - Table

GeneralColumnsConstraintsAdvancedPartitionsParametersSecuritySQL

Primary KeyForeign KeyCheckUniqueExclude

NameColumns

emp\_mgr\_ekmgr

GeneralDefinitionColumns

Columns

Columnmgr

| Column         | Operator class                       | DESC           | NULLs order      | Operator     |
|----------------|--------------------------------------|----------------|------------------|--------------|
| <div>mgr</div> | <div>Select the operator class</div> | <div>ASC</div> | <div>FIRST</div> | <div>=</div> |

Include columns

Select the column(s)

i?

CancelResetSave

Use the fields in the **Columns** tab to specify the column(s) to which the constraint applies. Use the drop-down listbox next to **Column** to select a column and click the **Add** icon (+) to provide details of the action on the column:

- The **Column** field is populated with the selection made in the **Column** drop-down listbox.
- If applicable, use the drop-down listbox in the **Operator class** to specify the operator class that will be used by the index for the column.
- Move the **DESC** switch to **DESC** to specify a descending sort order. The default is **ASC** which specifies an ascending sort order.
- Move the **NULLs order** switch to **LAST** to define an ascending sort order for NULLs. The default is **FIRST** which specifies a descending order.
- Use the drop-down list next to **Operator** to specify a comparison or conditional operator.

Click the **Advanced** tab to continue.

**Create - Table**

General Columns **Advanced** Constraints Partitions Parameters Security SQL

RLS Policy? ☐ No

Force RLS Policy? ☐ No

Of type

Fill factor

Toast tuple target

Parallel workers

Has OIDs? ☐ No

Unlogged? ☐ No

**Like**

Relation

With default values? ☐ No

With constraints? ☐ No

With indexes? ☐ No

With storage? ☐ No

With comments? ☐ No

Use the fields in the **Advanced** tab to define advanced features for the table:

- Move the **RLS Policy?** switch to the **Yes** position to enable the Row Level Security.
- Move the **Force RLS Policy?** to the **Yes** position to force the policy on the owner of the table.
- Use the drop-down listbox next to **Of type** to copy the table structure from the specified composite type. Please note that a typed table will be dropped if the type is dropped (with DROP TYPE ... CASCADE).
- Use the **Fill Factor** field to specify a fill factor for the table. The fill factor for a table is a percentage between 10 and 100. 100 (complete packing) is the default.
- Use the **Toast tuple target** field to set toast\_tuple\_target storage parameter of the table. The toast\_tuple\_target value is in bytes and has minimum value of 128. This field will be enabled only for PostgreSQL version >= 11
- Use the **Parallel workers** field to set parallel\_workers storage parameter of the table. The parallel\_workers sets the number of workers that should be used to assist a parallel scan of the table.
- Move the **Has OIDs?** switch to the **Yes** position to specify that each row within a table has a system-assigned object identifier. The default is **No**.
- Move the **Unlogged?** switch to the **Yes** position to disable logging for the table. Data written to an unlogged table is not written to the write-ahead log. Any indexes created on an unlogged table are automatically unlogged as well. The default is **No**.

Use the fields in the **Like** box to specify which attributes of an existing table from which a table will automatically copy column names, data types, and not-null constraints; after saving the new or modified table, any changes to the original table will not be applied to the new table.

- Use the drop-down listbox next to **Relation** to select a reference table.
- Move the **With default values?** switch to the **Yes** position to copy default values.
- Move the **With constraints?** switch to the **Yes** position to copy table and column constraints.
- Move the **With indexes?** switch to the **Yes** position to copy indexes.
- Move the **With storage?** switch to the **Yes** position to copy storage settings.
- Move the **With comments?** switch to the **Yes** position to copy comments.

With PostgreSQL 10 forward, the **Partition** tab will be visible.

Click the **Partition** tab to continue.



**Create - Table**

General Columns Constraints Advanced **Partitions** Parameters Security SQL

Partition Type: Range

**Partition Keys**

| Key type | Column | Expression |
|----------|--------|------------|
|----------|--------|------------|

Partition table supports two types of keys:  
**Column:** User can select any column from the list of available columns.  
**Expression:** User can specify expression to create partition key.  
**Example:** Let's say, we want to create a partition table based per year for the column 'saledate', having datatype 'date/timestamp', then we need to specify the expression as 'extract(YEAR from saledate)' as partition key.

**Partitions**

| Operation | Name | Default | From | To | In | Modulus | Remainder |
|-----------|------|---------|------|----|----|---------|-----------|
|-----------|------|---------|------|----|----|---------|-----------|

**Create a table:** User can create multiple partitions while creating new partitioned table. Operation switch is disabled in this scenario.  
**Edit existing table:** User can create/attach/detach multiple partitions. In attach operation user can select table from the list of suitable tables to be attached.  
**Default:** The default partition can store rows that do not fall into any existing partition's range or list.  
**From/To/In input:** From/To/In input: Values for these fields must be quoted with single quote. For more than one partition key values must be comma(,) separated.  
**Example: From/To:** Enabled for range partition. Consider partitioned table with multiple keys of type Integer, then values should be specified like '100','200'.  
**In:** Enabled for list partition. Values must be comma(,) separated and quoted with single quote.  
**Modulus/Remainder:** Enabled for hash partition.

Buttons: ? i Cancel Reset Save

Use the fields in the **partition** tab to create the partitions for the table:

- Select a partition type from the **Partition Type** selection box. There are 3 options available; Range, List and Hash. Hash option will only enable for PostgreSQL version >= 11.

Use the **Partition Keys** panel to define the partition keys. Click the **Add** icon (+) to add each partition keys selection:

- Select a partition key type in the **Keytype** field.
- Select a partition column in the **Column** field if Column option selected for **Keytype** field .
- Specify the expression in the **Expression** field if Expression option selected for the **Keytype** field.

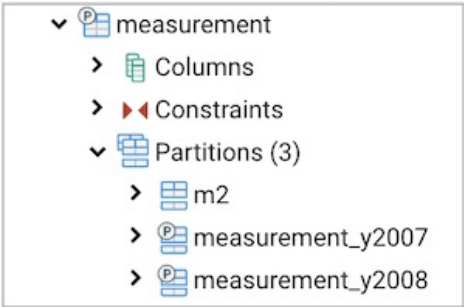
Use the **Partitions** panel to define the partitions of a table. Click the **Add** icon (+) to add each partition:

- Move the **Operation** switch to **attach** to attach the partition, by default it is **create** .
- Use the **Name** field to add the name of the partition.
- If partition type is Range or List then **Default** field will be enabled.
- If partition type is Range then **From** and **To** fields will be enabled.
- If partition type is List then **In** field will be enabled.
- If partition type is Hash then **Modulus** and **Remainder** fields will be enabled.

Users can create a partition and define them as a partitioned table. Click the **Edit** icon to expand the properties of a partition. Use the **Partition** tab to create that partition as a partitioned table.

- Move the **Partitioned Table?** switch to the **Yes** in case you want to create a partitioned table.
- Select a partition type from the **Partition Type** selection box.
- Use the **Partition Keys** panel to define the partition keys.

View of multi level Partitioned Table in browser tree:



Click the **Parameter** tab to continue.

Create - Table

GeneralColumnsConstraintsAdvancedPartitionsParametersSecuritySQL

TableTOAST table

Custom auto-vacuum?

No

Autovacuum Enabled?

✓ Not set

Yes

No

| Label                  | Value | Default     |
|------------------------|-------|-------------|
| ANALYZE scale factor   |       | 0.10        |
| ANALYZE base threshold |       | 50          |
| FREEZE maximum age     |       | 200,000,000 |
| VACUUM cost delay      |       | 20          |
| VACUUM cost limit      |       | -1          |
| VACUUM scale factor    |       | 0.20        |
| VACUUM base threshold  |       | 50          |
| FREEZE minimum age     |       | 50,000,000  |
| FREEZE table age       |       | 150,000,000 |

i?

✕ Cancel

↺ Reset

💾 Save

Use the tabs nested inside the **Parameter** tab to specify VACUUM and ANALYZE thresholds; use the **Table** tab and the **Toast Table** tab to customize values for the table and the associated toast table:

- Move the **Custom auto-vacuum?** switch to the **Yes** position to perform custom maintenance on the table and to select values in the **Vacuum table**. The **Vacuum Table** provides default values for maintenance operations.
- Changing **Autovacuum enabled?** to **Not set** will reset autovacuum\_enabled.

Provide a custom value in the **Value** column for each metric listed in the **Label** column.

Click the **Security** tab to continue.

Create - Table

GeneralColumnsConstraintsAdvancedPartitionsParametersSecuritySQL

Privileges

GranteePrivilegesGrantor

Security labels

ProviderSecurity label

Cancel

Reset

Save

Use the **Security** tab to assign privileges and define security labels.

Use the **Privileges** panel to assign privileges to a role. Click the **Add** icon (+) to set privileges for database objects:

- Select the name of the role from the drop-down listbox in the **Grantee** field.
- Click inside the **Privileges** field. Check the boxes to the left of one or more privileges to grant the selected privilege to the specified user.
- The current user, who is the default grantor for granting the privilege, is displayed in the **Grantor** field.

Click the **Add** icon (+) to assign additional privileges; to discard a privilege, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Use the **Security Labels** panel to define security labels applied to the function. Click the **Add** icon (+) to add each security label selection:

- Specify a security label provider in the **Provider** field. The named provider must be loaded and must consent to the proposed labeling operation.
- Specify a security label in the **Security Label** field. The meaning of a given label is at the discretion of the label provider. PostgreSQL places no restrictions on whether or how a label provider must interpret security labels; it merely provides a mechanism for storing them.

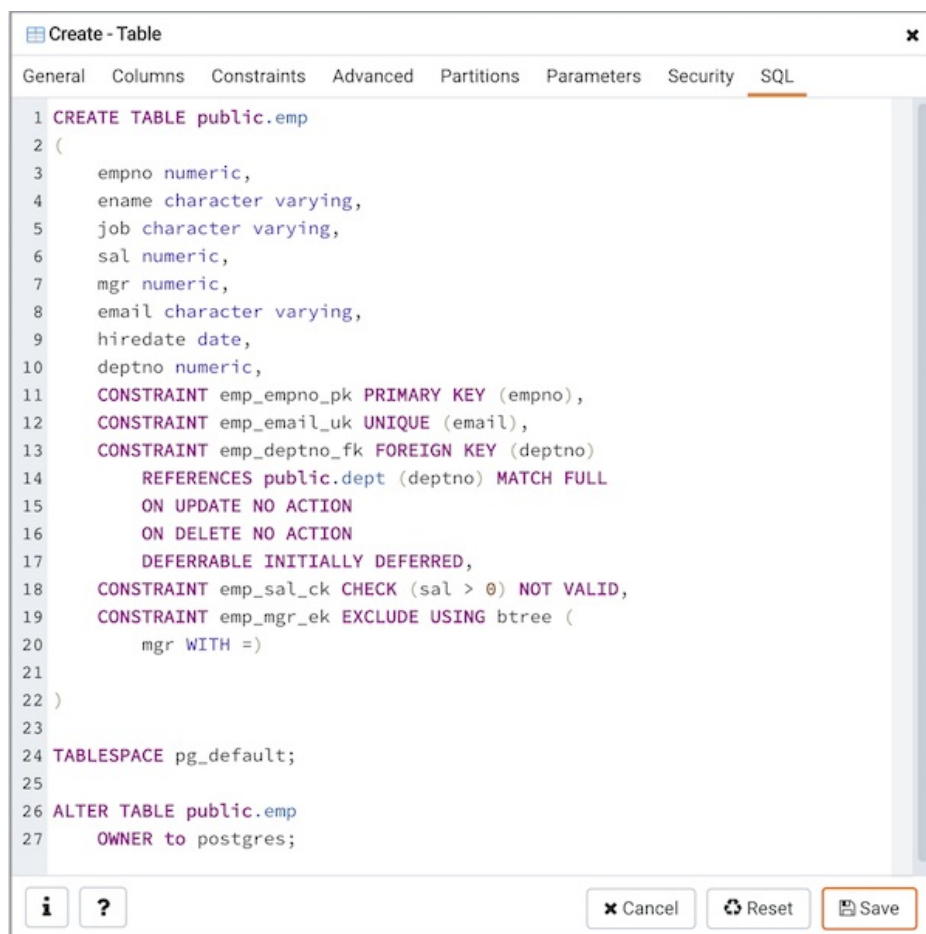
Click the **Add** icon (+) to assign additional security labels; to discard a security label, click the trash icon to the left of the row and confirm deletion in the **Delete Row** popup.

Click the **SQL** tab to continue.

Your entries in the **Table** dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

Example

The following is an example of the sql command generated by user selections in the **Table** dialog:



The example shown demonstrates creating a table named `product_category`. It has three columns and a primary key constraint on the `category_id` column.

- Click the **Info** button (i) to access online help. View context-sensitive help in the **Tabbed browser**, where a new tab displays the PostgreSQL core documentation.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

### 34.5.12.11 Trigger Dialog

Use the **Trigger** dialog to create a trigger or modify an existing trigger. A trigger executes a specified function when certain events occur.

The **Trigger** dialog organizes the development of a trigger through the following dialog tabs: **General**, **Definition**, **Events**, and **Code**. The **SQL** tab displays the SQL code generated by dialog selections.

→ Create - Trigger

General Definition Events Transition Code SQL

Name log\_delete

Comment

i ? Cancel Reset Save

Use the fields in the **General** tab to identify the trigger:

- Use the **Name** field to add a descriptive name for the trigger. This must be distinct from the name of any other trigger for the same table. The name will be displayed in the **Browser** tree control. Note that if multiple triggers of the same kind are defined for the same event, they will be fired in alphabetical order by name.
- Store notes about the trigger in the **Comment** field.

Click the **Definition** tab to continue.

→ Create - Trigger

General

Definition

Events

Transition

Code

SQL

Row trigger?

Yes

Constraint trigger?

No

Deferrable?

No

Deferred?

No

Trigger function

public.fki\_cascade\_del

Arguments

i

?

Cancel

Reset

Save

Use the fields in the **Definition** tab to define the trigger:

- Move the **Row trigger?** switch to the **No** position to disassociate the trigger from firing on each row in a table. The default is **Yes**.
- Move the **Constraint trigger?** switch to the **Yes** position to specify the trigger is a constraint trigger.
- If enabled, move the **Deferrable?** switch to the **Yes** position to specify the timing of the constraint trigger is deferrable and can be postponed until the end of the statement. The default is **No**.
- If enabled, move the **Deferred?** switch to the **Yes** position to specify the timing of the constraint trigger is deferred to the end of the statement causing the triggering event. The default is **No**.
- Use the drop-down listbox next to **Trigger Function** to select a trigger function or procedure.
- Use the **Arguments** field to provide an optional (comma-separated) list of arguments to the function when the trigger is executed. The arguments are literal string constants.

→ log\_delete

General

Definition

Events

Transition

Code

SQL

Trigger enabled?

Enable

Row trigger?

Yes

Constraint trigger?

No

Deferrable?

No

Deferred?

No

Trigger function

public.fki\_cascade\_del

Arguments

i

?

Cancel

Reset

Save

- **Trigger enabled** field is available in trigger dialog once the trigger is created. You can select one of the four options available.

Click the **Events** tab to continue.

**Create - Trigger**

General Definition **Events** Transition Code SQL

Fires BEFORE

**Events**

INSERT No UPDATE Yes

DELETE Yes TRUNCATE No

When 1

Columns x deptno x

**i ?** Cancel Reset Save

Use the fields in the **Events** tab to specify how and when the trigger fires:

- Use the drop-down listbox next to the **Fires** fields to determine if the trigger fires **BEFORE** or **AFTER** a specified event. The default is **BEFORE**.
- Select the type of event(s) that will invoke the trigger; to select an event type, move the switch next to the event to the **YES** position. The supported event types are **INSERT**, **UPDATE**, **DELETE**, and **TRUNCATE**.
- Use the **When** field to provide a boolean condition that will invoke the trigger.
- If defining a column-specific trigger, use the **Columns** field to specify the columns or columns that are the target of the trigger.

Click the **Code** tab to continue.



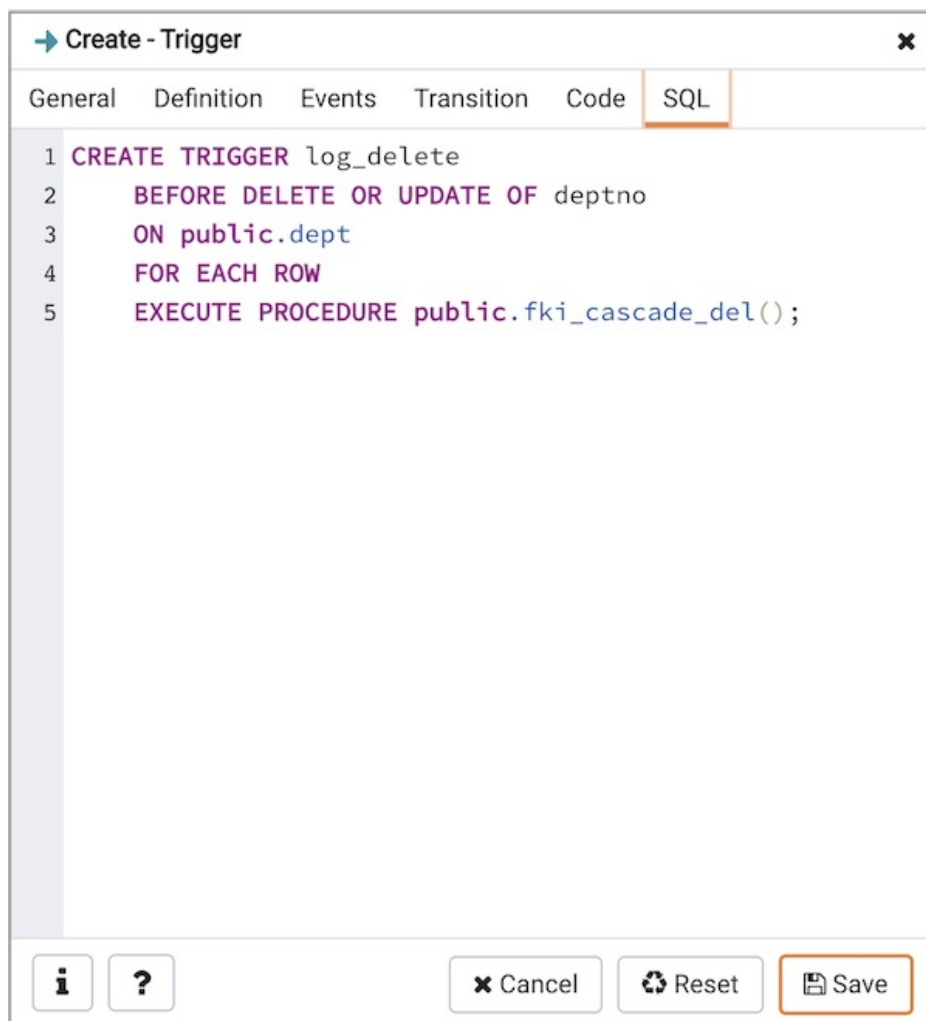
Use the **Code** field to specify any additional code that will be invoked when the trigger fires.

Click the **SQL** tab to continue.

Your entries in the **Trigger** dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

#### Example

The following is an example of the sql command generated by user selections in the **Trigger** dialog:



The example demonstrates creating a trigger named `log_update` that calls a procedure named `log_account_update` that logs any updates to the `distributors` table.

- Click the **Info** button (i) to access online help. View context-sensitive help in the **Tabbed browser**, where a new tab displays the PostgreSQL core documentation.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

### 34.5.12.12 Unique Constraint Dialog

Use the **Unique constraint** dialog to define a unique constraint for a specified table. Unique constraints ensure that the data contained in a column, or a group of columns, is unique among all the rows in the table.

The **Unique constraint** dialog organizes the development of a unique constraint through the following dialog tabs: **General** and **Definition**. The **SQL** tab displays the SQL code generated by dialog selections.

Create - Unique constraint

General

Definition

SQL

Name

dept\_loc\_uk

Comment

i

?

Cancel

Reset

Save

Use the fields in the **General** tab to identify the unique constraint:

- Use the **Name** field to add a descriptive name for the unique constraint. The name will be displayed in the **Browser** tree control.

Click the **Definition** tab to continue.

Use the fields in the **Definition** tab to define the unique constraint:

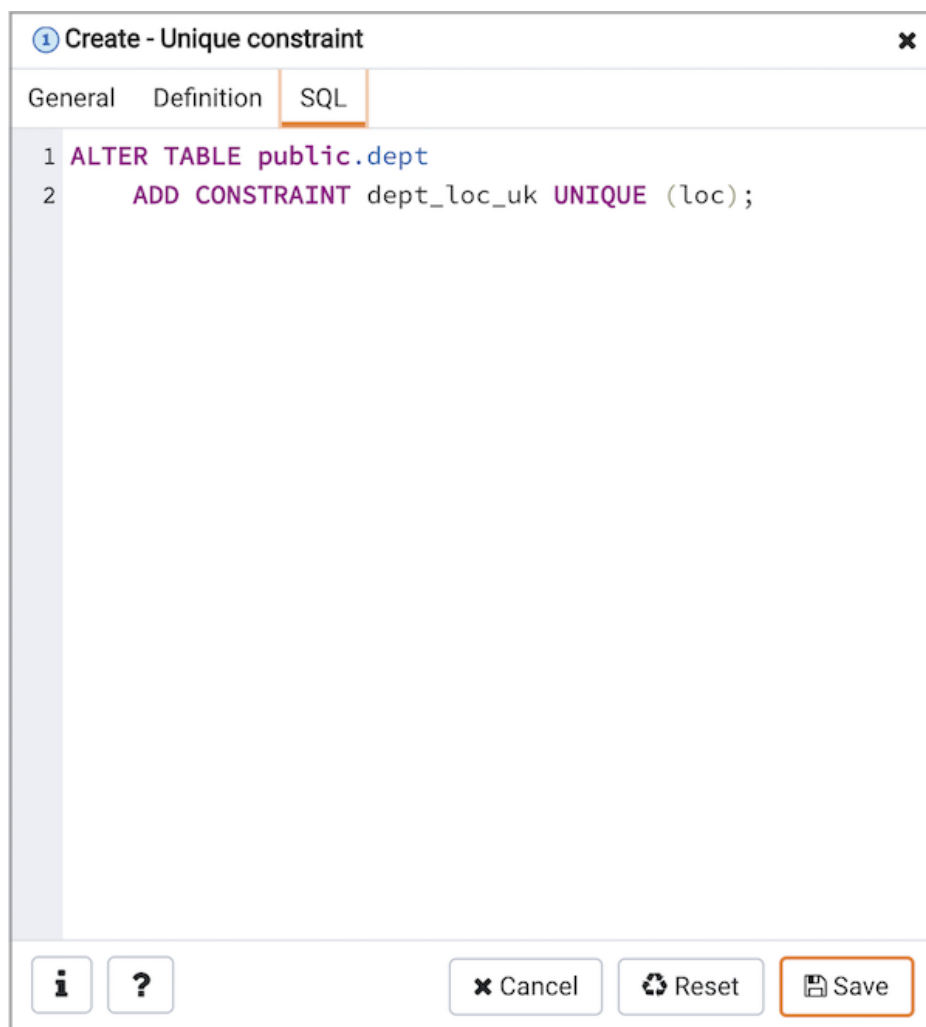
- Click inside the **Columns** field and select one or more column names from the drop-down listbox. To delete a selection, click the **x** to the left of the column name. The unique constraint should be different from the primary key constraint defined for the same table; the selected column(s) for the constraints must be distinct.
- Use **Include columns** field to specify columns for **INCLUDE** clause of the constraint. This option is available in Postgres 11 and later.
- Select the name of the tablespace in which the unique constraint will reside from the drop-down listbox in the **Tablespace** field.
- Select the name of an index from the drop-down listbox in the **Index** field. This field is optional. Adding a unique constraint will automatically create a unique B-tree index on the column or group of columns listed in the constraint, and will force the column(s) to be marked NOT NULL.
- Use the **Fill Factor** field to specify a fill factor for the table and index. The fill factor for a table is a percentage between 10 and 100. 100 (complete packing) is the default.
- Move the **Deferrable?** switch to the **Yes** position to specify the timing of the constraint is deferrable and can be postponed until the end of the statement. The default is **No**.
- If enabled, move the **Deferred?** switch to the **Yes** position to specify the timing of the constraint is deferred to the end of the statement. The default is **No**.

Click the **SQL** tab to continue.

Your entries in the **Unique constraint** dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

#### Example

The following is an example of the sql command generated by user selections in the **Unique constraint** dialog:



The example shown demonstrates creating a unique constraint named `name_con` on the `name` column of the `distributors` table.

- Click the **Info** button (i) to access online help. View context-sensitive help in the **Tabbed browser**, where a new tab displays the PostgreSQL core documentation.
- Click the **Save** button to save work.
- Click the **Cancel** button to exit without saving work.
- Click the **Reset** button to restore configuration parameters.

## 34.6 Monitoring Barman

Postgres Enterprise Manager (PEM) is designed to assist database administrators, system architects, and performance analysts when administering, monitoring, and tuning PostgreSQL and Advanced Server database servers.

Barman (Backup and Recovery Manager) is an open-source administration tool for remote backups and disaster recovery of PostgreSQL servers in business-critical environments. It relies on PostgreSQL's Point In Time Recovery technology, allowing DBAs to remotely manage a complete catalogue of backups and the recovery phase of multiple remote servers – all from one location. For more information about Barman, see [Barman docs](#).

Starting with version 8.4, you can monitor a Barman server through PEM console. You can monitor your Barman server using the PEM console.

Before you manage a Barman server through PEM console, your system must meet certain requirements. See

- [Prerequisites for monitoring Barman](#)

You must add a Barman server to the PEM console whose backup you want to manage with Barman. See

- [Configuring a Barman Server](#)

After you configure the Barman server, you will be able to view the details of the backups in the dashboard. See

- [Viewing the Barman Server Details on a PEM Dashboard](#)

### 34.6.1 Prerequisites for managing Barman

Before adding a Barman server to the PEM console, install the following on the Barman host:

- [Barman](#)
- [Postgres backup API](#)

### 34.6.2 Configuring a Barman Server

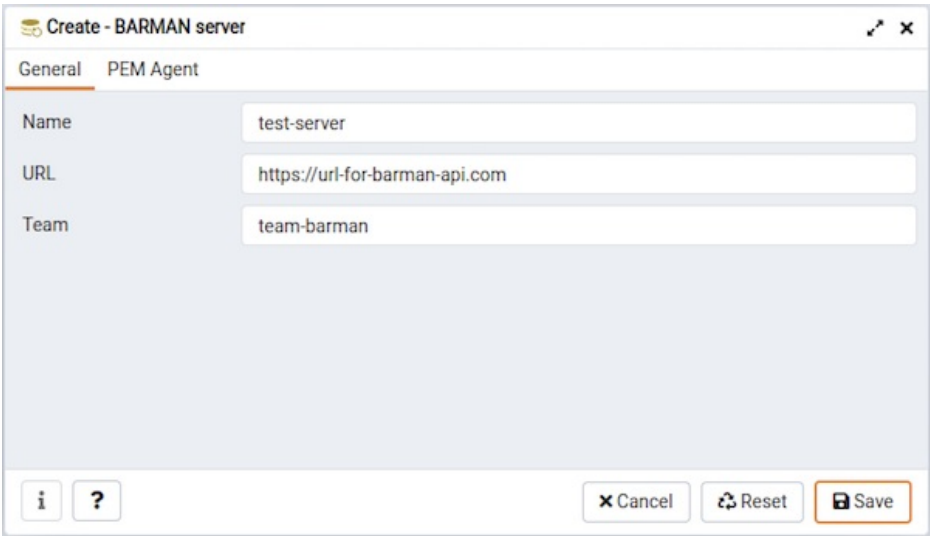
You can configure and edit your Barman server using:

- PEM web client
- `pemworker` command line

#### Using PEM web client

##### Configure

You can use the `Create-BARMAN server` dialog to register an existing Barman server with the PEM server. To access the dialog, right-click on the `BARMAN Servers` node and select `Create-BARMAN Server`.



Use the fields on the `General` tab to describe the general properties of the Barman server:

- Use the `Name` field to specify a user-friendly name for the server. The name identifies the server in the browser tree.

- Use the **URL** field to specify the URL of the host where Barman is installed.
- Use the **Team** field to specify a PostgreSQL role name. Only PEM users who are members of this role, who created the server initially, or have superuser privileges on the PEM server will see this server when they log on to PEM. If this field is left blank, all PEM users see the server.

Create - BARMAN server

General

PEM Agent

Bound Agent

Postgres Enterprise Manager Host

Probe frequency

300

Probe execution frequency in seconds

Heartbeat

10

Heartbeat interval in seconds

i

?

Cancel

Reset

Save

Use the fields on the **PEM Agent** tab to specify connection details for the PEM Agent:

- Use the **Bound Agent** field to select the agent that you want to configure as a Barman server. Only those PEM agents that are supported for Barman are listed in the drop-down list.
- Use the **Probe Frequency** field to specify the number of seconds to execute the probes with the specified interval.
- Use the **Heartbeat** field to specify the number of seconds interval to check the availability of PEM agent.

Note

After registering the **Barman server** you need to restart the PEM agent.

Editing

To edit your Barman server, select your Barman server from the browser tree, right click and select **Properties**.

test-5

General

PEM Agent

Information

Configurations

Name

test-5

URL

http://test.com

Team

i

?

Cancel

Reset

Save

- Use the fields on the PEM Agent tab to modify the **Bound Agent** , **Probe Frequency** , and **Heartbeat** . Only the owner of the Barman server can modify the fields on PEM Agent tab.

test-5

General

PEM Agent

Information

Configurations

| Parameter | Value |
|-----------|-------|
|-----------|-------|

i

?

Cancel

Reset

Save

- Use the fields on Information tab to view the detailed information about your Barman server. This tab gets populated whenever the Barman related probes are executed. You cannot modify any of the fields on the Information tab.

test-5

General

PEM Agent

Information

Configurations

| Parameter                     | Value                      |
|-------------------------------|----------------------------|
| backup_method                 | rsync                      |
| backup_options                | concurrent_backup          |
| barman_home                   | /var/lib/barman            |
| barman_user                   | barman                     |
| basebackup_retry_sleep        | 30                         |
| basebackup_retry_times        | 3                          |
| compression                   | pigz                       |
| configuration_files_directory | /etc/barman.d              |
| errors_list                   | []                         |
| immediate_checkpoint          | False                      |
| last_backup_maximum_age       | 1 WEEK                     |
| log_file                      | /var/log/barman/barman.log |
| minimum_redundancy            | 3                          |
| network_compression           | False                      |
| parallel_jobs                 | 1                          |
| path_prefix                   | /usr/lib/postgresql/11/bin |
| recovery_options              | get-wal                    |
| retention_policy              | RECOVERY WINDOW OF 4 WEEKS |
| reuse_backup                  | link                       |

i

?

Cancel

Reset

Save

- Use the fields on Configuration tab to view the configuration settings of your Barman server. This tab gets populated whenever the Barman related probes are executed. You cannot modify any of the fields on the Configuration tab.

Note



After registering the **Barman server** you need to restart the PEM agent.

## Using **pemworker** command line

You can configure Barman server using **pemworker** command line options.

```
asheshvashi@pem:~/PEM/agent$ ./pemworker --update-barman --help
./pemworker --update-barman [barman-update-options]

barman-update-options:
  --id <barman-id> (ID for the existing BARMAN API 'pg-backup-api')
  --api-url <url> (URL of the BARMAN API 'pg-backup-api')
  --probe-execution-frequency <interval> (Default: 30, Probe the BARMAN API 'pg-backup-api' at regular
interval 'in seconds' and fetch the metrics.)
  --heartbeat-interval <interval> (Default: 10, Ping the BARMAN API 'pg-backup-api' 'status' API at a
regular interval 'in seconds' for checking its availability.)
  --ssl-crt <certificate_file> (SSL certificate file for the BARMAN API.)
  --ssl-key <key_file> (Private SSL key for the BARMAN API.)
  --ssl-ca-cert <ca_file> (CA certificate to verify peer against the BARMAN API.)
  --config-file/-c <config_file> (Path to the agent configuration file.)

asheshvashi@pem:~/PEM/agent$ ./pemworker --unregister-barman --help
./pemworker --unregister-barman [barman-unregistration-options]

barman-unregistration-options:
  --id <barman-id> (ID for the existing BARMAN API, registered with the PEM Server.'pg-backup-api')
  --config-file/-c <config_file> (Path to the agent configuration file.)

asheshvashi@pem:~/PEM/agent$ ./pemworker --register-barman --help
./pemworker --register-barman [barman-registration-options]

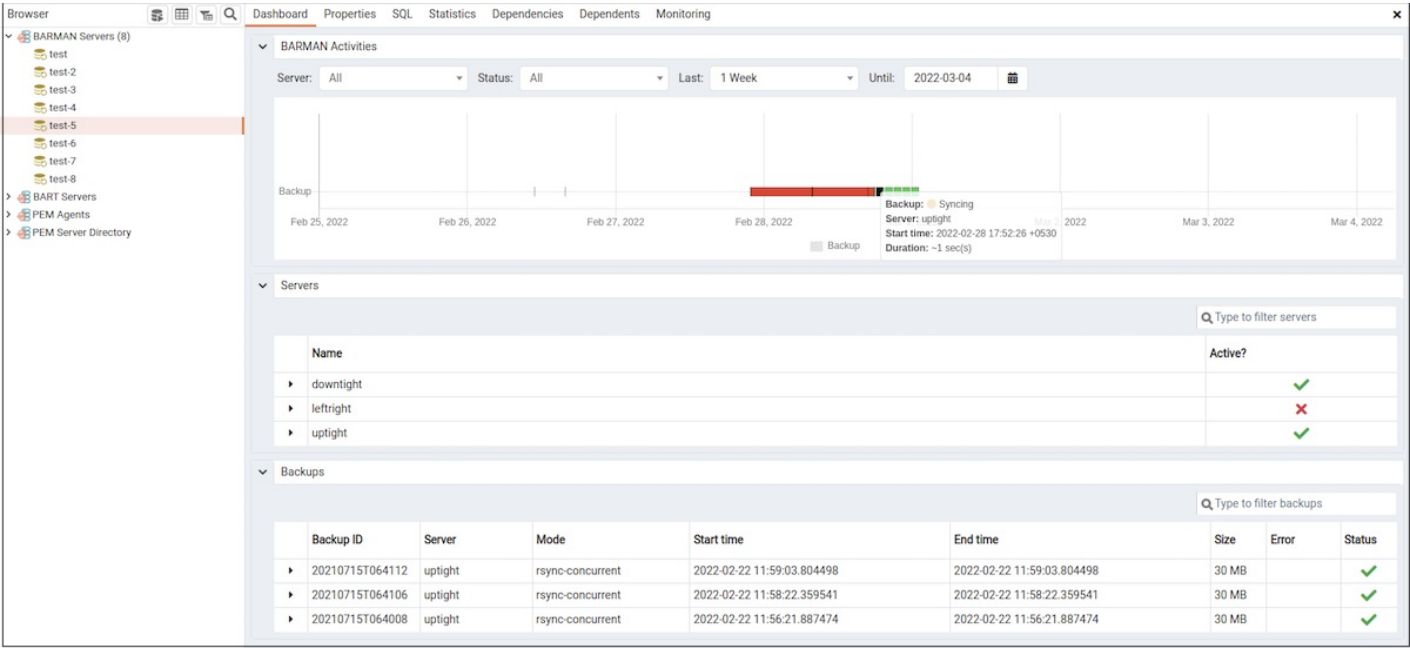
barman-registration-options:
  --api-url <url> (URL of the BARMAN API 'pg-backup-api')
  --description <name> (Description to show on the UI 'User interface' for the BARMAN API.)
  --probe-execution-frequency <interval> (Default: 30, Probe the BARMAN API 'pg-backup-api' at regular
interval 'in seconds' and fetch the metrics.)
  --heartbeat-interval <interval> (Default: 10, Ping the BARMAN API 'pg-backup-api' 'status' API at a
regular interval 'in seconds' for checking its availability.)
  --ssl-crt <certificate_file> (SSL certificate file for the BARMAN API.)
  --ssl-key <key_file> (Private SSL key for the BARMAN API.)
  --ssl-ca-cert <ca_file> (CA certificate to verify peer against the BARMAN API.)
  --team <database-role> (Specify the name of the database group role, on the PEM backend database
server, that should have access to this BARMAN API Server.)
  --owner <database-user> (Specify the name of the database user, on the PEM backend database server,
who will own the BARMAN API Server.)
  --config-file/-c <config_file> (Path to the agent configuration file.)
```

### Note

After registering the **Barman server** you need to restart the PEM agent.

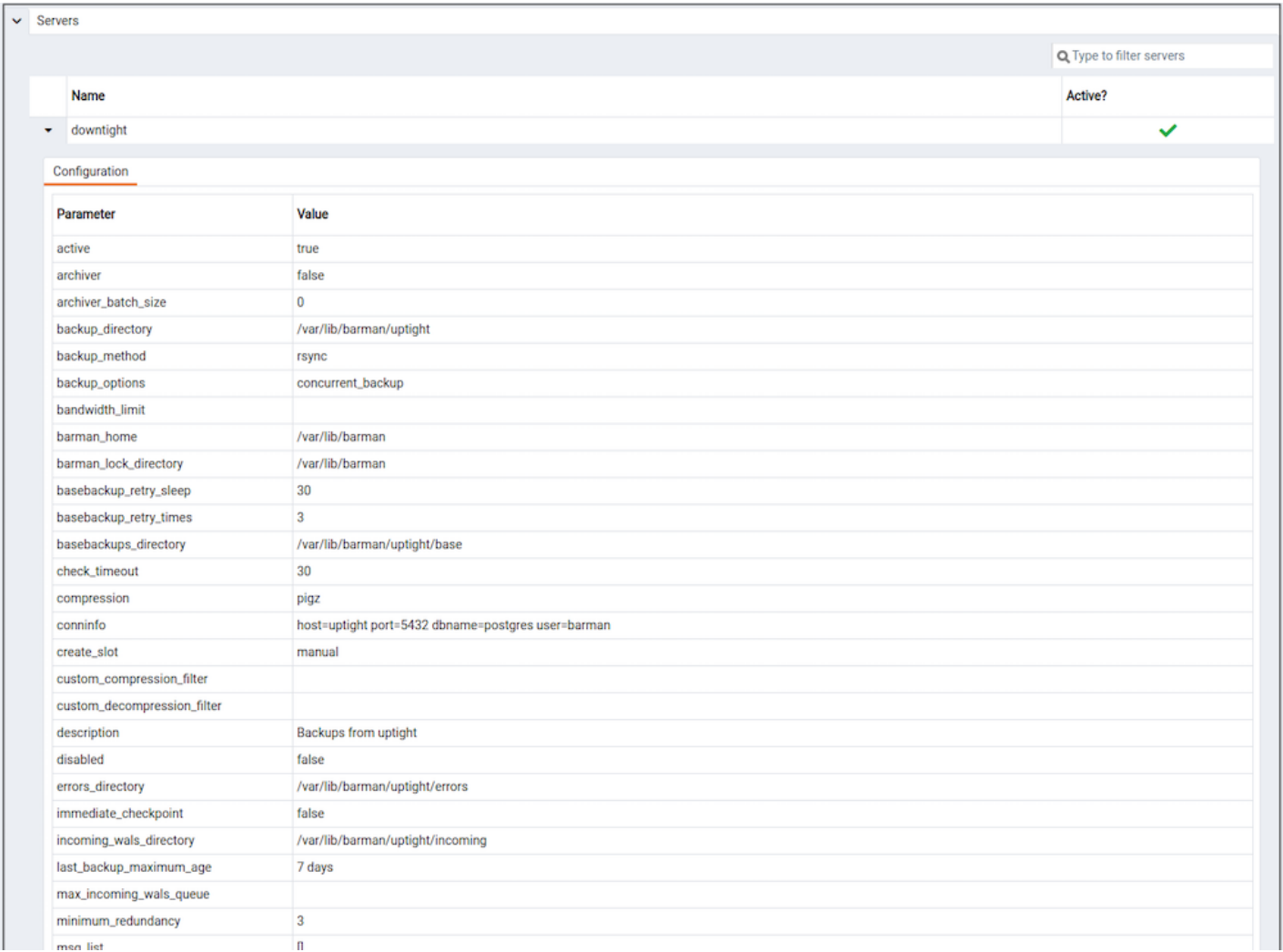
## 34.6.3 Viewing the Barman Server details on a PEM dashboard

Once the Barman server is configured, you can see the entire backup- and server-related details for that particular Barman server on the PEM Dashboard.



When you select a monitored Barman server, details of all the associated database servers along with their activities are displayed as a chart on the Dashboard in the **Barman Activities** panel. You can select the activities on any criteria that you specify in the filter boxes (the database server, status, duration, or date).

The **Servers** panel displays a list of all the database servers managed by that particular Barman server along with the active status.



|                               |  |
|-------------------------------|--|
| name                          | uptight  |
| network_compression           | false  |
| parallel_jobs                 | 1  |
| path_prefix                   | /usr/lib/postgresql/11/bin                                   |
| post_archive_retry_script     |  |
| post_archive_script           |  |
| post_backup_retry_script      |  |
| post_backup_script            |  |
| post_delete_retry_script      |  |
| post_delete_script            |  |
| post_recovery_retry_script    |  |
| post_recovery_script          |  |
| post_wal_delete_retry_script  |  |
| post_wal_delete_script        |  |
| pre_archive_retry_script      |  |
| pre_archive_script            |  |
| pre_backup_retry_script       |  |
| pre_backup_script             |  |
| pre_delete_retry_script       |  |
| pre_delete_script             |  |
| pre_recovery_retry_script     |  |
| pre_recovery_script           |  |
| pre_wal_delete_retry_script   |  |
| pre_wal_delete_script         |  |
| primary_ssh_command           |  |
| recovery_options              | get-wal  |
| retention_policy              | window 4 w   |
| retention_policy_mode         | auto   |
| reuse_backup                  | link   |
| slot_name                     | backup_unarmed   |
| ssh_command                   | ssh -q postgres@uptight                                      |
| streaming_archiver            | true   |
| streaming_archiver_batch_size | 0  |
| streaming_archiver_name       | barman_receive_wal   |
| streaming_backup_name         | barman_streaming_backup                                      |
| streaming_conninfo            | host=uptight port=5432 dbname=postgres user=streaming_barman |
| streaming_wals_directory      | /var/lib/barman/uptight/streaming                            |
| tablespace_bandwidth_limit    |  |
| wal_retention_policy          | simple-wal 4 w   |
| wals_directory                | /var/lib/barman/uptight/wals                                 |

leftright
uptight

✗
✓

The **Backups** panel displays a list of all the backups of the database servers managed by that particular Barman server. You can filter the list to display the details of a particular database server. You can also filter the list on any criteria that you specify in the filter box. Typically, this filter works with any kind of string value (excluding date, time, and size) listed under the columns. For example, you can type **tar** to filter the list and display only those backups that are in tar format.

Backup details includes the **Backup ID**, **Server**, **Mode**, **Start time**, **End time**, **Size**, **Error**, and **Status** column.

| Backups                  |  |                  |                            |                            |       |       |        |  |
|--------------------------|--|------------------|----------------------------|----------------------------|-------|-------|--------|--|
| Q Type to filter backups |  |                  |                            |                            |       |       |        |  |
| Backup ID                | Server   | Mode             | Start time                 | End time                   | Size  | Error | Status |  |
| 20210715T064112          | uptight  | rsync-concurrent | 2022-02-22 11:59:03.804498 | 2022-02-22 11:59:03.804498 | 30 MB |       | ✓      |  |
| Backup details           |  |                  |                            |                            |       |       |        |  |
| Parameter                | Value  |                  |                            |                            |       |       |        |  |
| backup_id                | 20210715T064112  |                  |                            |                            |       |       |        |  |
| backup_label             | START WAL LOCATION: 0/D000028 (file 000000010000000000000000) \nCHECKPOINT LOCATION: 0/D000060 \nBACKUP METHOD: streamed \nBACKUP FROM: master \n                                |                  |                            |                            |       |       |        |  |
| begin_offset             | 40   |                  |                            |                            |       |       |        |  |
| begin_time               | Thu Jul 15 06:41:12 2021   |                  |                            |                            |       |       |        |  |
| begin_wal                | 000000010000000000000000   |                  |                            |                            |       |       |        |  |
| begin_xlog               | 0/D000028  |                  |                            |                            |       |       |        |  |
| config_file              | /opt/postgres/data/postgresql.conf   |                  |                            |                            |       |       |        |  |
| copy_stats               | { "analysis_time": 0.497281, "analysis_time_per_item": { "pgdata": 0.497281 }, "copy_time": 0.482371, "copy_time_per_item": { "pg_control": 0.148338, "pgdata": 0.332809 }, "... |                  |                            |                            |       |       |        |  |
| deduplicated_size        | 41992  |                  |                            |                            |       |       |        |  |
| end_offset               | 248  |                  |                            |                            |       |       |        |  |
| end_time                 | Thu Jul 15 06:41:13 2021   |                  |                            |                            |       |       |        |  |
| end_wal                  | 000000010000000000000000   |                  |                            |                            |       |       |        |  |
| end_xlog                 | 0/D0000F8  |                  |                            |                            |       |       |        |  |
| error                    |  |                  |                            |                            |       |       |        |  |
| hba_file                 | /opt/postgres/data/pg_hba.conf   |                  |                            |                            |       |       |        |  |
| ident_file               | /opt/postgres/data/pg_ident.conf   |                  |                            |                            |       |       |        |  |
| included_files           | /opt/postgres/data/conf.d/0000-tpa.conf, /opt/postgres/data/conf.d/0001-tpa_restart.conf, /opt/postgres/data/conf.d/1111-extensions.conf, /opt/postgres/data/conf....            |                  |                            |                            |       |       |        |  |
| mode                     | rsync-concurrent   |                  |                            |                            |       |       |        |  |
| pgdata                   | /opt/postgres/data   |                  |                            |                            |       |       |        |  |
| server_name              | uptight  |                  |                            |                            |       |       |        |  |
| size                     | 31975345   |                  |                            |                            |       |       |        |  |
| status                   | DONE   |                  |                            |                            |       |       |        |  |
| systemid                 | 6985025709346482144  |                  |                            |                            |       |       |        |  |
| tablespaces              |  |                  |                            |                            |       |       |        |  |
| timeline                 | 1  |                  |                            |                            |       |       |        |  |
| version                  | 110012   |                  |                            |                            |       |       |        |  |
| xlog_segment_size        | 16777216   |                  |                            |                            |       |       |        |  |
| 20210715T064106          | uptight  | rsync-concurrent | 2022-02-22 11:58:22.359541 | 2022-02-22 11:58:22.359541 | 30 MB |       | ✓      |  |
| 20210715T064008          | uptight  | rsync-concurrent | 2022-02-22 11:56:21.887474 | 2022-02-22 11:56:21.887474 | 30 MB |       | ✓      |  |

## 34.7 SQL Profiler

SQL Profiler captures statistical information and query execution plans for SQL statements executed during a trace session. You can use the information stored by SQL Profiler to identify performance issues. Before using SQL Profiler, you must [install and configure SQL Profiler](#) on each database you intend to profile.

### 34.7.1 Installing SQL Profiler

SQL Profiler allows a database superuser to locate and optimize poorly-running SQL code. Users of Microsoft SQL Server's Profiler will find PEM's SQL Profiler very similar in operation and capabilities. SQL Profiler is installed with each Advanced Server instance; if you are using PostgreSQL, you must download the SQL Profiler installer or packages and install the SQL Profiler product into each managed database instance you wish to profile.

SQL Profiler is officially supported only on the EnterpriseDB distributions of PostgreSQL version 9.4 or above and Advanced Server version 9.4 or above. The plugin is distributed via StackBuilder, or is available from the [EnterpriseDB website](#)

You can use the graphical installer to install any version of SQL Profiler on Windows platform.

On Linux, if you have installed your database server through graphical installer then you must use the graphical installer to install the SQL Profiler. If you have installed your database server using the RPM or DEB package, then you must use the RPM or DEB package to install the SQL Profiler.

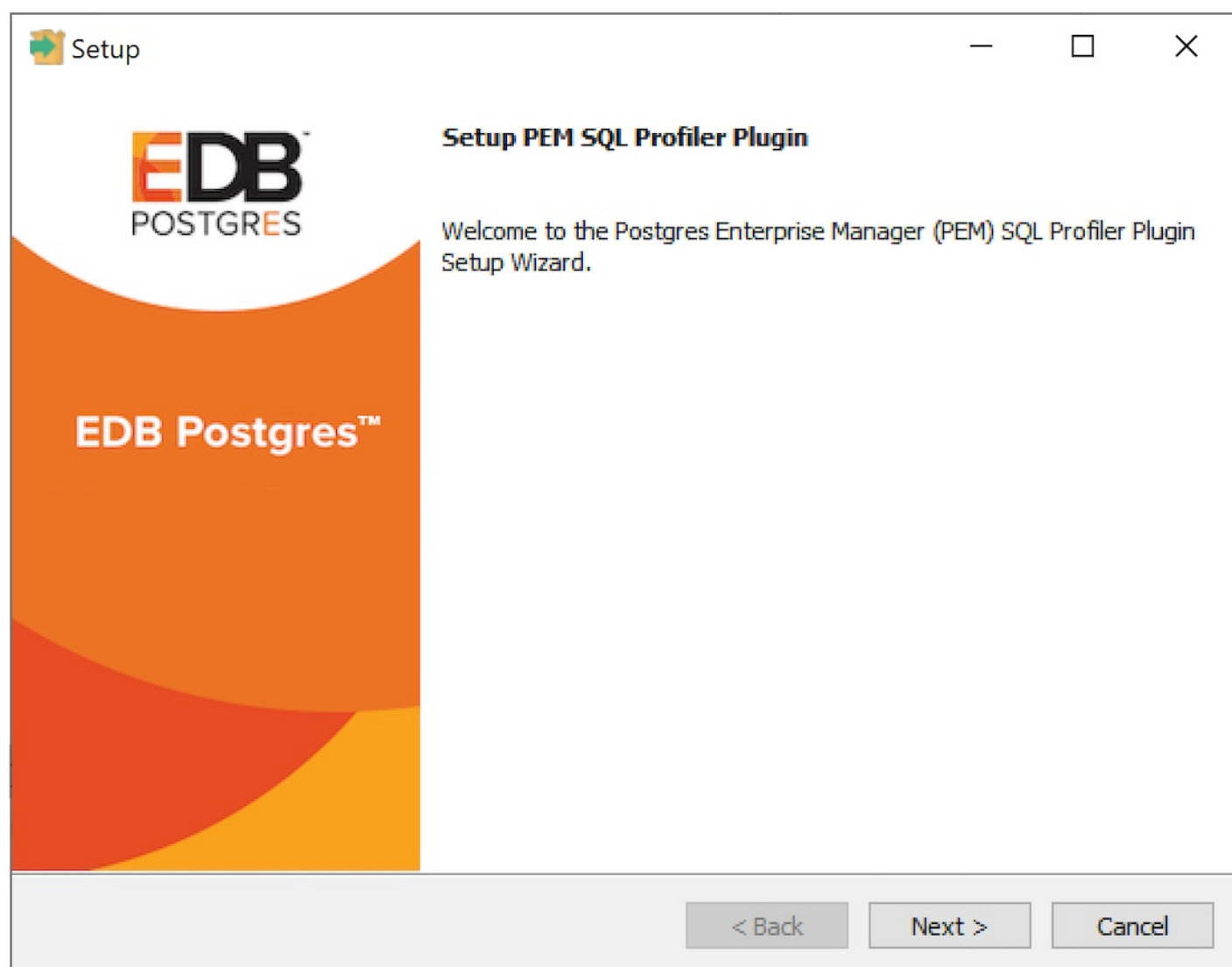
## Installing SQL Profiler on Windows

To invoke the SQL Profiler graphical installer, assume superuser privileges (or `Administrator` privileges on Windows), navigate into the directory that contains the installer, and invoke the installer:

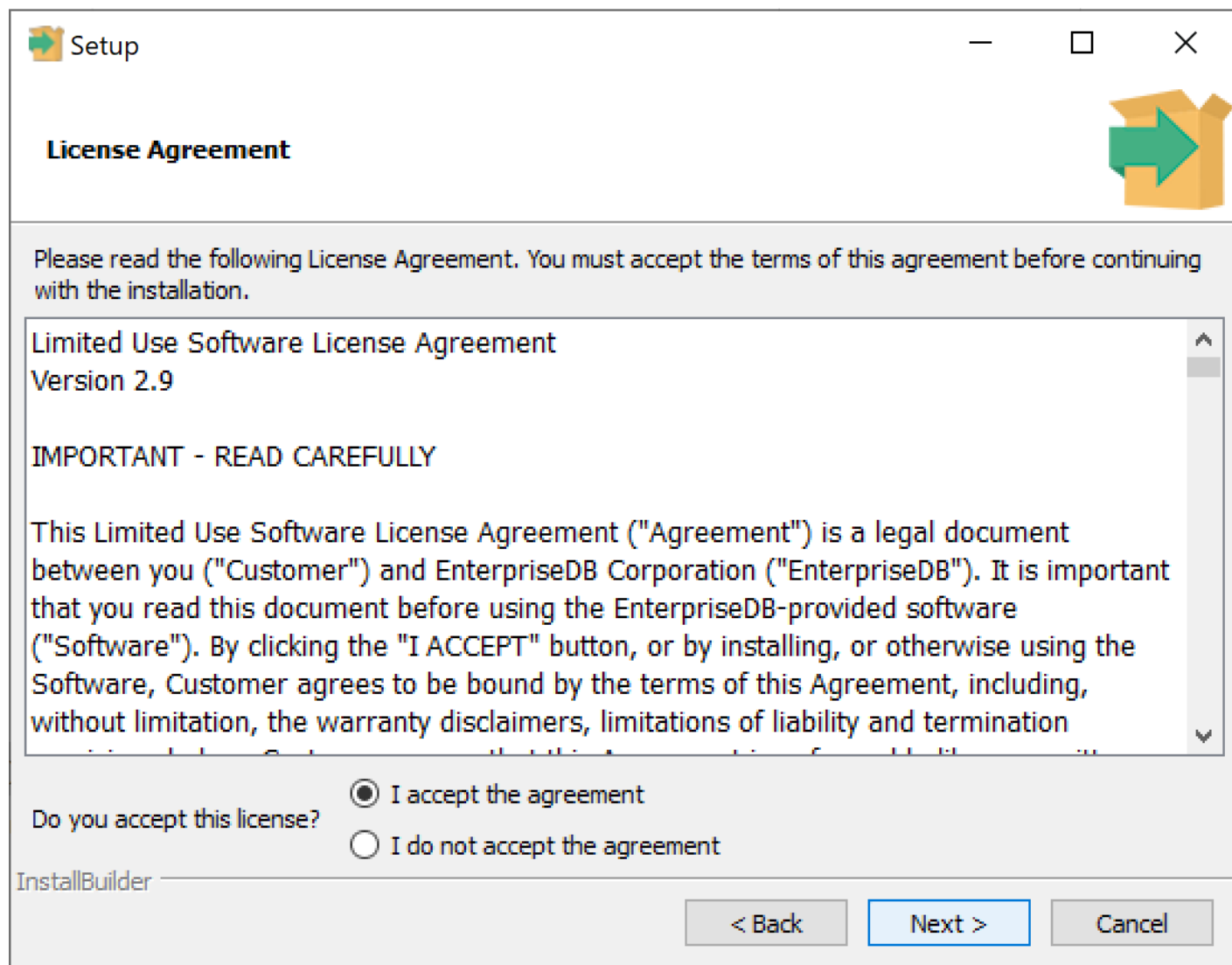
```
sqlprofiler-pg-<pg_version>-<sql_profiler_version>-windows-x64.exe
```

Where, `pg_version` is the version of your PostgreSQL and `sql_profiler_version` is the version of SQL Profiler.

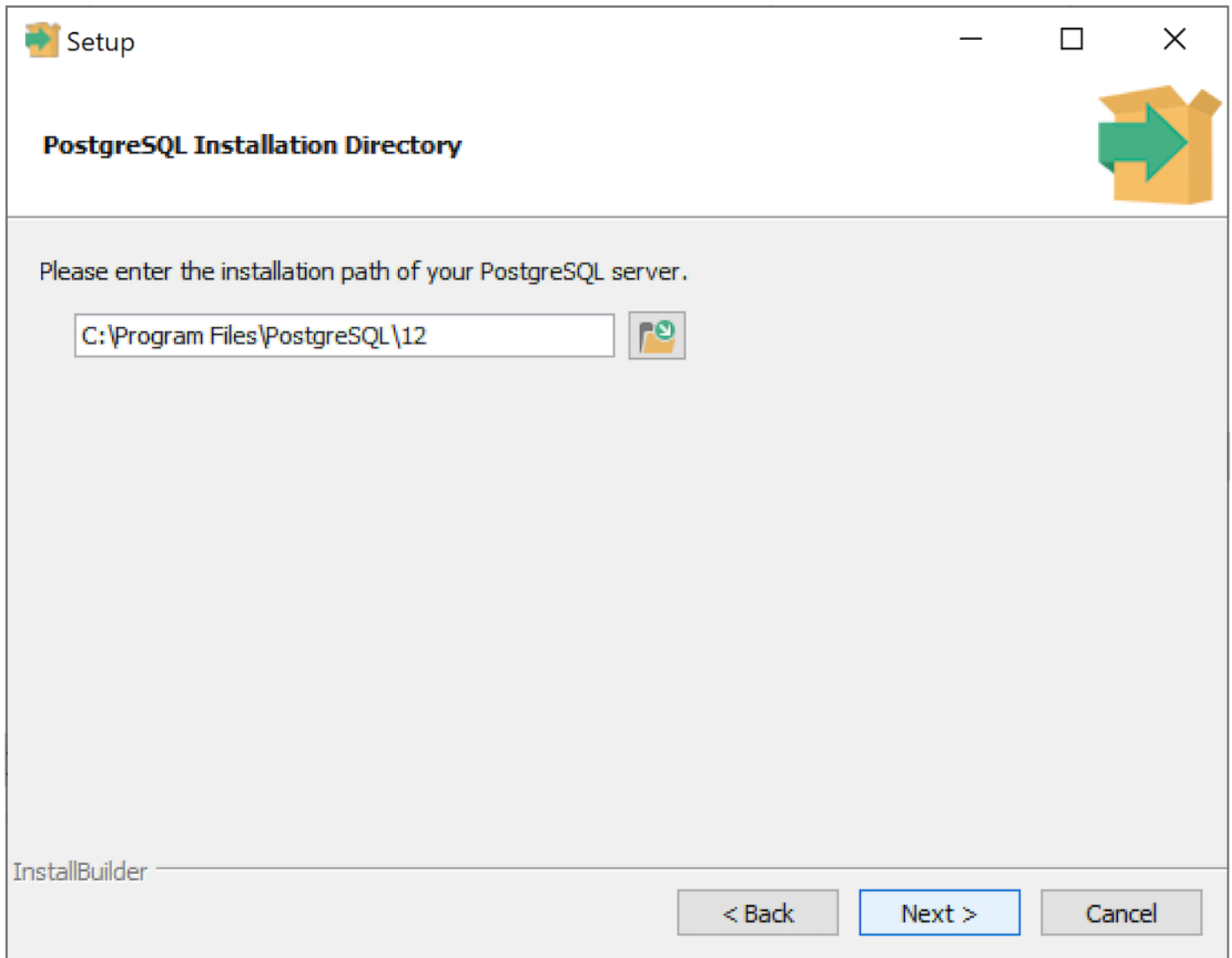
The SQL Profiler installer welcomes you to the Setup Wizard.



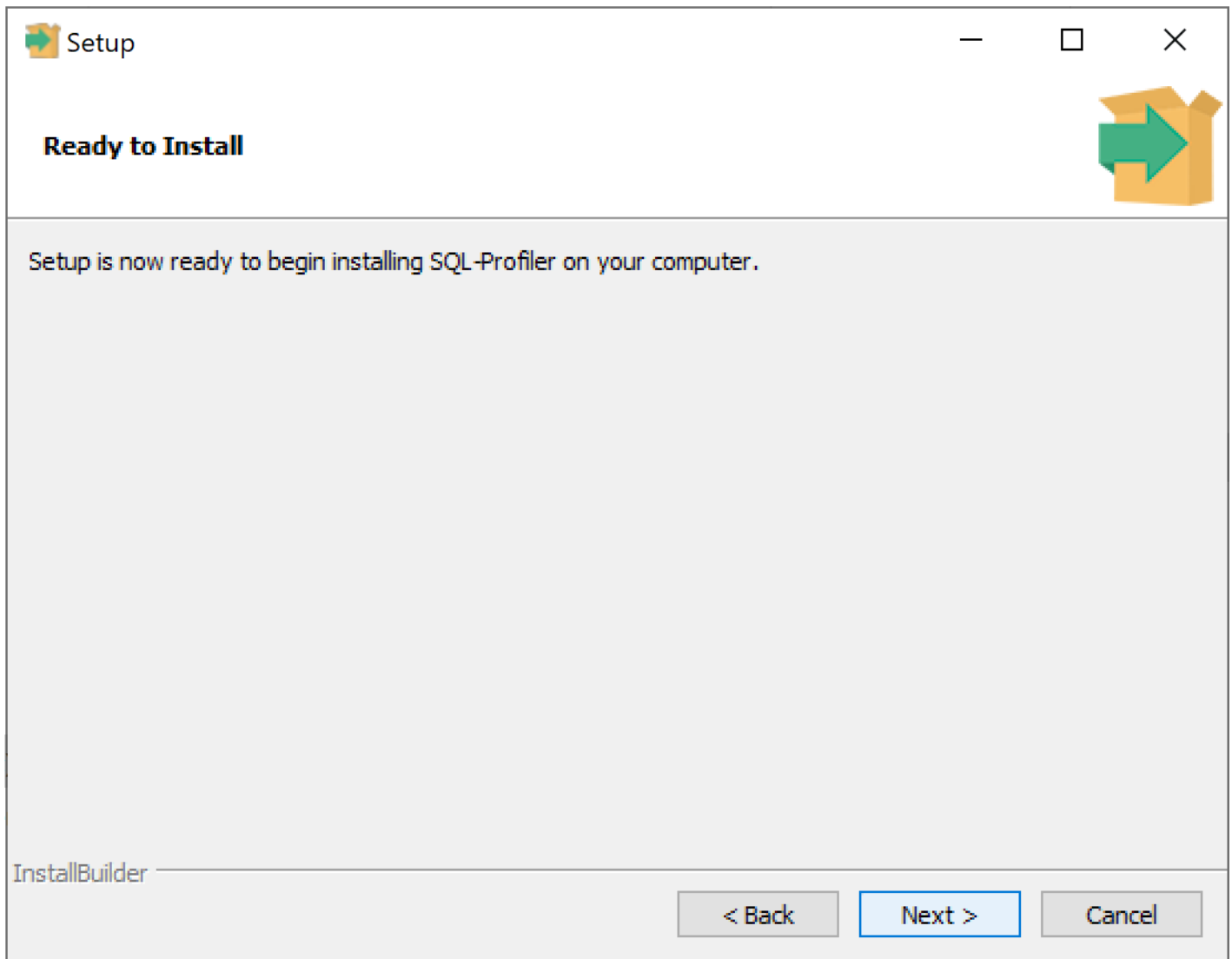
Click `Next` to continue to the `License Agreement`.



Carefully review the license agreement before highlighting the appropriate radio button and accepting the agreement; click **Next** to continue to the **Installation Directory** dialog.

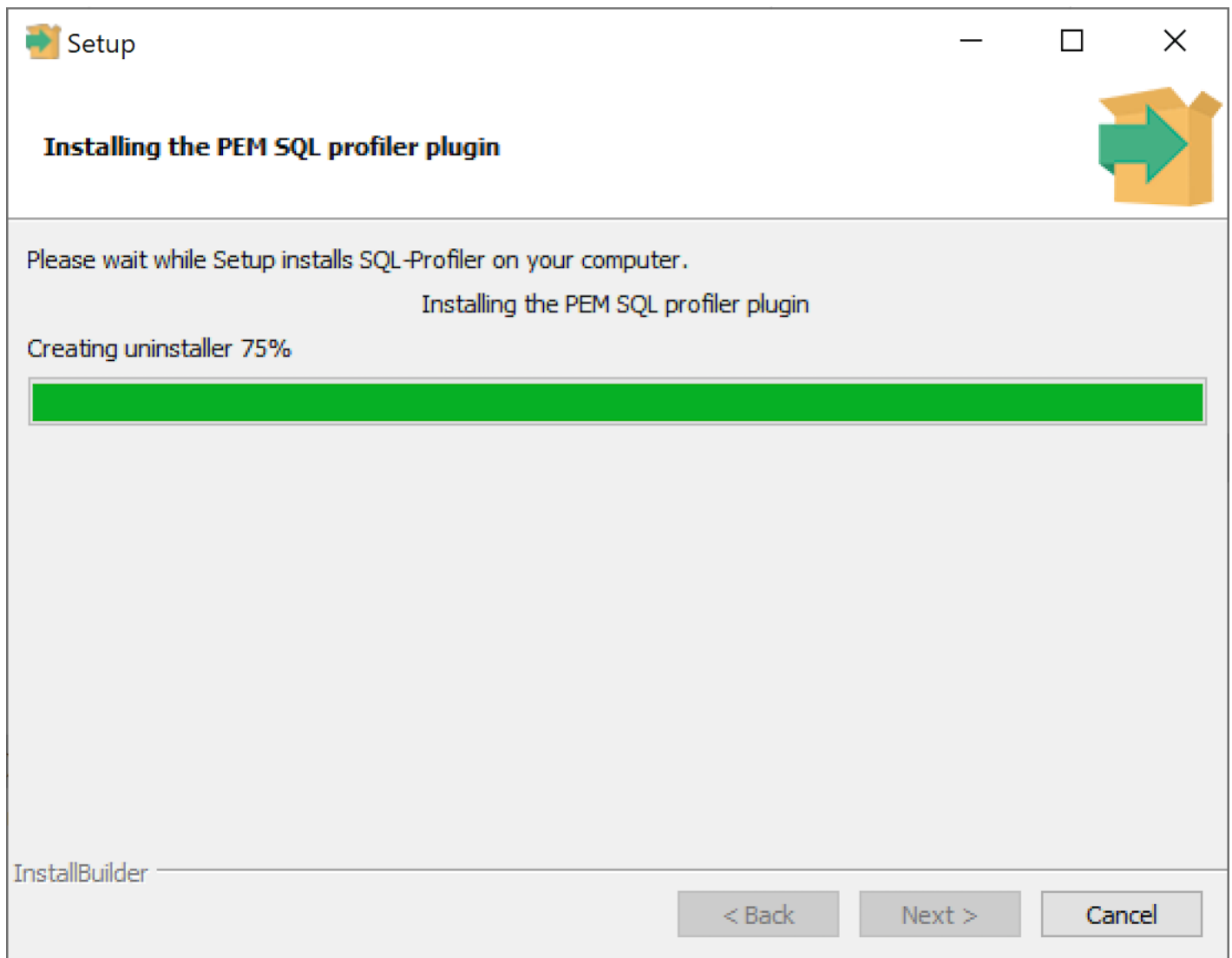


Specify an alternate location for the installation directory, or accept the default location and click **Next** to continue.

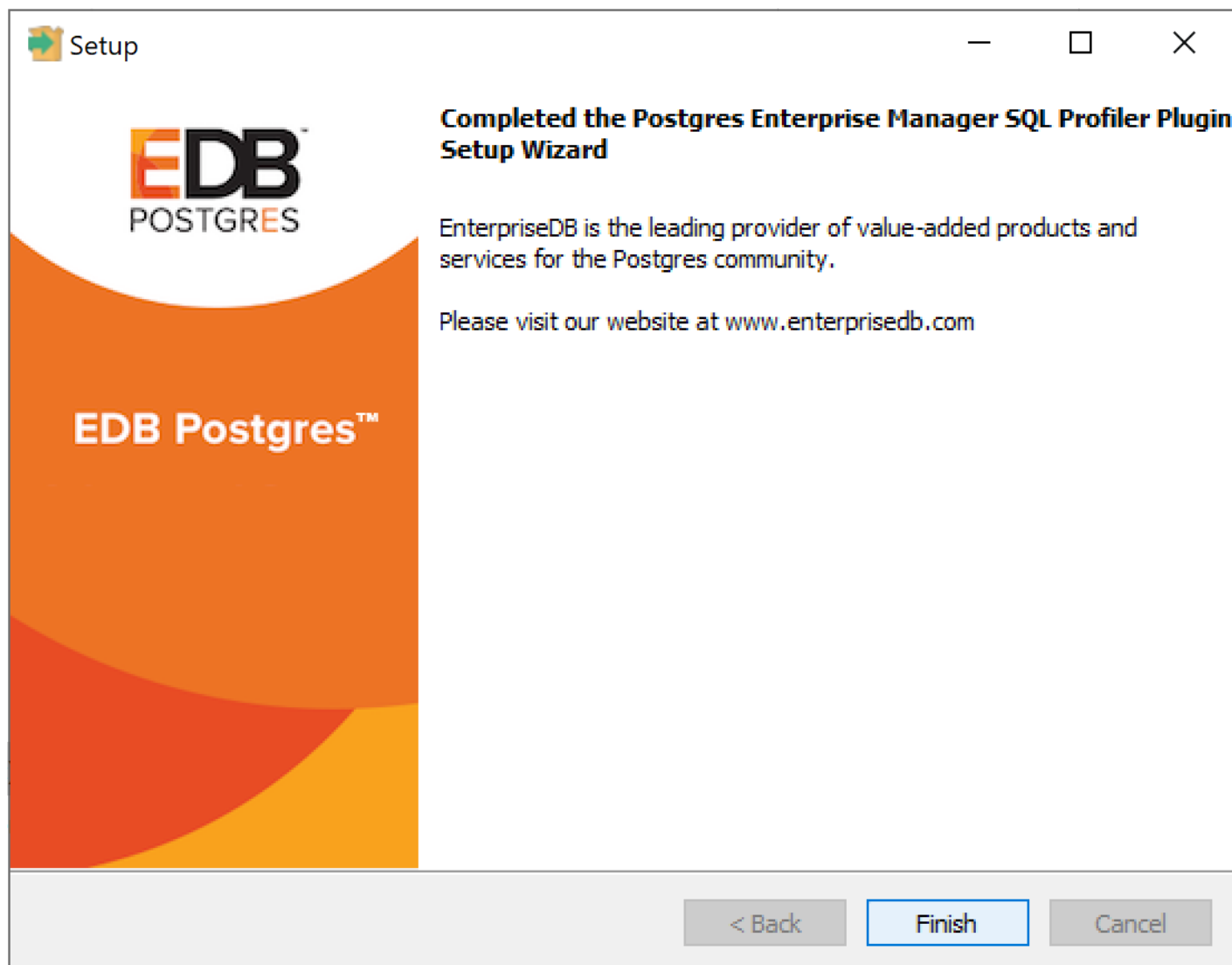


The wizard is now ready to install the SQL Profiler plugin. Click [Next](#) to continue.





The SQL Profiler plugin installer displays progress bars as it copies files to your system.



When the installation is complete, the SQL Profiler plugin is ready to be configured.

### Installing SQL Profiler on Linux using RPMs

#### Note

You may be required to add the `ssltutils` package to your PostgreSQL database servers before installing SQL Profiler.

You can install SQL Profiler using rpm on RHEL or Centos 6 or 7, using yum command as root user:

```
yum install postgresql<pg_version>-sqlprofiler
```

Where, `pg_version` is the version of your PostgreSQL.

When the installation is complete, the SQL Profiler plugin is ready to be configured.

### Installing SQL Profiler on Debian/Ubuntu using DEB

**Note**

You may be required to add the `sslutils` package to your PostgreSQL database servers before installing SQL Profiler.

You can install SQL Profiler using DEB on Ubuntu 18, using apt command as root user:

```
apt install postgresql-<pg_version>-sqlprofiler
```

Where, `pg_version` is the version of your PostgreSQL.

When the installation is complete, the SQL Profiler plugin is ready to be configured.

## 34.7.2 Configuring SQL Profiler

The SQL Profiler plugin is not automatically enabled when the installation process completes. This allows you to restart the server at a convenient time, and prevents the plugin from being loaded unnecessarily on systems where it is not required on a continual basis.

Use the following steps to enable the plugin for each database monitored by SQL Profiler:

1. Edit the `postgresql.conf` file on the server you wish to profile, modifying the `shared_preload_libraries` configuration parameter as shown below:

For Linux, `shared_preload_libraries = '$libdir/sql-profiler'`

For Windows, `shared_preload_libraries = '$libdir/sql-profiler.dll'`

2. Create the functions used by SQL Profiler in your database. The SQL Profiler installation program places a SQL script (named `sql-profiler.sql`) in the `share/contrib` subdirectory of the main PostgreSQL installation directory. You must invoke this script on the maintenance database (specified when registering the server with PEM).

You can also use the psql command line to invoke the configuration script. The following command uses psql to invoke the `sql-profiler.sql` script on PostgreSQL Server database on a Linux system:

```
$ /usr/pgsql-x/bin/psql -U postgres postgres -f /usr/pgsql-x/share/contrib/sql-profiler.sql
```

3. Stop and re-start the server for the changes to take effect.

Please note: if you have connected to the PEM server with the PEM client before configuring SQL Profiler, you must disconnect and reconnect with the server to enable SQL Profiler functionality.

## 34.7.3 Using SQL Profiler

You can use SQL Profiler to create and store up to 15 named traces; use menu options to create and manage traces.

### Creating a Trace

You can use the Create trace... dialog to define a SQL Trace for any database on which SQL Profiler has been installed and configured. [installed and configured](#). To access the dialog, highlight the name of the database in the PEM client tree control; navigate through the **Management** menu to the **SQL Profiler** pull-aside menu, and select **Create trace...**

**Create trace...**

Trace options | Schedule | Periodic job options

Name: test\_trace

User filter: enterisedb

Database filter: postgres

Maximum trace file size(MB): 0  
Note: 0(Zero) indicates maximum trace file size to be 250 MB

Run now? Yes

? Cancel Create

Use the fields on the **Trace options** tab to specify details about the new trace:

- Provide a name for the trace in the **Name** field.
- Click in the **User filter** field to specify the roles whose queries will be included the trace; optionally, check the box next to **Select All** to include queries from all roles.
- Click in the **Database filter** field to specify which databases to trace; optionally, check the box next to **Select All** to include queries against all databases.
- Specify a trace size in the **Maximum Trace File Size** field; SQL Profiler will terminate the trace when it reaches approximately the size specified.
- Specify **Yes** in the **Run Now** field to start the trace when you select the **Create** button; select **No** to enable fields on the **Schedule** tab.

**Create trace...**

Trace options | Schedule | Periodic job options

Start time:

End time:

Repeat? No  
This option will allow you to schedule periodic trace job.

? Cancel Create

Use the fields on the **Schedule** tab to specify scheduling details for the new trace:

- Use the **Start time** field to specify the starting time for the trace.
- Use the **End time** field to specify the ending time for the trace.
- Specify **Yes** in the **Repeat?** field to indicate that the trace should be repeated every day at the times specified; select **No** to enable fields on the **Periodic job options** tab.

Create trace...

Trace options

Schedule

Periodic job options

Days

Week days

Month days

Months

Times

Hours

Minutes

Schedules are specified using a cron-style format.

• For each selected time or date element, the schedule will execute.  
e.g. To execute at 5 minutes past every hour, simply select '05' in the Minutes list box.

• Values from more than one field may be specified in order to further control the schedule.  
e.g. To execute at 12:05 and 14:05 every Monday and Thursday, you would click minute 05, hours 12 and 14, and weekdays Monday and Thursday.

• For additional flexibility, the Month days check list includes an extra Last day option. This matches the last day of the month, whether it happens to be the 28th, 29th, 30th or 31st.

?

Cancel

Create

Fields on the **Periodic job options** tab specify scheduling details for a recurring trace. Use fields in the **Days** section to specify the days on which the job will execute:

- Click in the **Week days** field to select the days of the week on which the trace will execute.
- Click in the **Month days** field to select the days of the month on which the trace will execute.
- Click in the **Months** field to select the months in which the trace will execute.

Use fields in the **Times** section to specify a time schedule for the trace execution:

- Click in the **Hours** field to select the hours at which the trace will execute.
- Click in the **Minutes** field to select the hours at which the trace will execute.

When you've completed the **Create trace...** dialog, click **Create** to start the newly defined trace or to schedule the trace for a later time. If you elect to execute the trace immediately, the trace results will display in the PEM client.

Dashboard Properties SQL Statistics Dependencies Dependents Monitoring SQL Profiler (Trace: test\_trace)

1 of 1

Show queries per page: 100

| #  | Start Time                    | Duration (ms) | Query                      | Rows Affected | User         | Database | PID   | File System Read | File System Write | Page Fault |
|----|-------------------------------|---------------|----------------------------|---------------|--------------|----------|-------|------------------|-------------------|------------|
| 1  | 2020-04-24 01:54:57.556323-07 | 1.273133      | SELECT heartbeat_inte...   | 1             | agent1       | pem      | 91666 | 48               | 0                 | 6          |
| 2  | 2020-04-24 01:54:57.627954-07 | 12.234771     | /*pga4dash*/ SELECT ...    | 5             | enterprisedb | postgres | 33485 | 0                | 0                 | 0          |
| 3  | 2020-04-24 01:54:57.647487-07 | 0.00245       | SELECT 1                   | 1             | enterprisedb | pem      | 30543 | 0                | 0                 | 0          |
| 4  | 2020-04-24 01:54:57.659666-07 | 0.004035      | SELECT pg_has_role(p...    | 1             | enterprisedb | pem      | 30543 | 0                | 0                 | 0          |
| 5  | 2020-04-24 01:54:57.660344-07 | 0.00083       | SELECT 1                   | 1             | enterprisedb | pem      | 30543 | 0                | 0                 | 0          |
| 6  | 2020-04-24 01:54:57.69393-07  | 0.917116      | SET DateStyle=ISO; SE...   | 1             | enterprisedb | postgres | 33900 | 0                | 0                 | 0          |
| 7  | 2020-04-24 01:54:57.696903-07 | 0.016561      | SELECT db.oid as did, ...  | 1             | enterprisedb | postgres | 33900 | 0                | 0                 | 0          |
| 8  | 2020-04-24 01:54:57.69824-07  | 0.027168      | SELECT oid as id, rolna... | 1             | enterprisedb | postgres | 33900 | 0                | 0                 | 0          |
| 9  | 2020-04-24 01:54:58.633668-07 | 12.019085     | /*pga4dash*/ SELECT ...    | 5             | enterprisedb | postgres | 33485 | 0                | 0                 | 0          |
| 10 | 2020-04-24 01:54:58.814973-07 | 0.001999      | SELECT 1                   | 1             | enterprisedb | pem      | 30543 | 0                | 0                 | 0          |
| 11 | 2020-04-24 01:54:58.826576-07 | 0.050517      | SELECT pg_has_role(p...    | 23            | enterprisedb | pem      | 30543 | 0                | 0                 | 0          |
| 12 | 2020-04-24 01:54:59.622713-07 | 11.296904     | /*pga4dash*/ SELECT ...    | 5             | enterprisedb | postgres | 33485 | 0                | 0                 | 0          |
| 13 | 2020-04-24 01:55:00.137269-07 | 0.001639      | SELECT 1                   | 1             | enterprisedb | postgres | 32578 | 0                | 0                 | 0          |
| 14 | 2020-04-24 01:55:01.250787-07 | 11.271025     | /*pga4dash*/ SELECT ...    | 5             | enterprisedb | postgres | 33485 | 0                | 0                 | 0          |
| 15 | 2020-04-24 01:55:02.114481-07 | 0.627253      | WITH agent_info AS (...)   | 1             | agent1       | pem      | 91666 | 680              | 0                 | 52         |
| 16 | 2020-04-24 01:55:02.236089-07 | 12.07968      | /*pga4dash*/ SELECT ...    | 5             | enterprisedb | postgres | 33485 | 0                | 0                 | 0          |
| 17 | 2020-04-24 01:55:02.628516-07 | 13.478983     | SELECT * FROM (SEL...      | 3             | agent1       | pem      | 91666 | 1568             | 0                 | 49         |
| 18 | 2020-04-24 01:55:03.232281-07 | 0.049985      | SELECT * FROM pem.j...     | 1             | agent1       | pem      | 91666 | 0                | 0                 | 0          |
| 19 | 2020-04-24 01:55:03.235696-07 | 0.011202      | SELECT nextval('pem.j...   | 1             | agent1       | pem      | 91666 | 0                | 0                 | 0          |
| 20 | 2020-04-24 01:55:03.239847-07 | 18.327522     | /*pga4dash*/ SELECT ...    | 5             | enterprisedb | postgres | 33485 | 0                | 0                 | 0          |
| 21 | 2020-04-24 01:55:03.243198-07 | 0.128994      | INSERT INTO pem.jobs...    | 1             | agent1       | pem      | 91666 | 32               | 0                 | 1          |
| 22 | 2020-04-24 01:55:03.263871-07 | 0.03972       | SELECT log_directory, L... | 1             | agent1       | pem      | 91666 | 0                | 0                 | 0          |

SQL Query Metrics

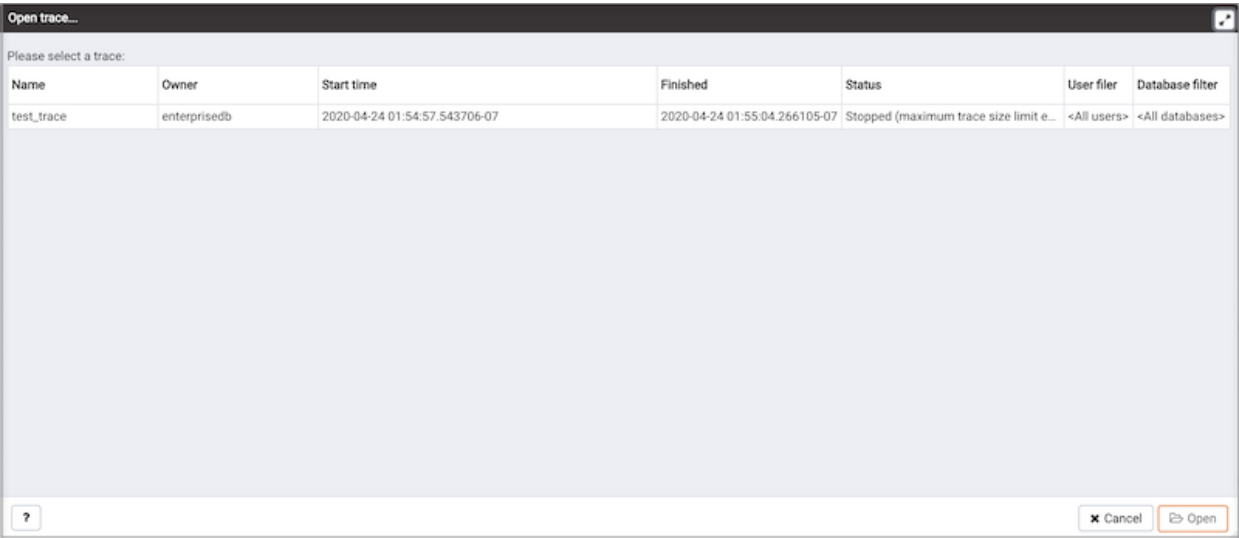
Graphical Plan Text-Based Plan

Graphical Analysis Statistics

Function Scan

Opening a Trace

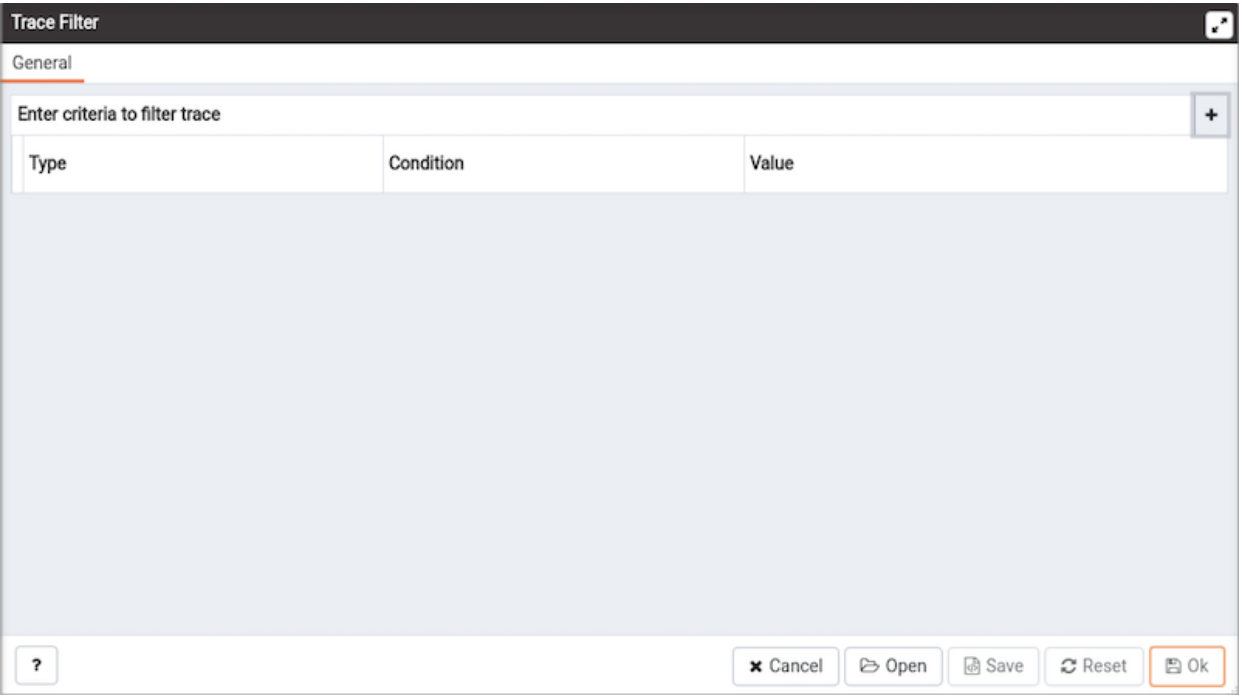
To view a previous trace, highlight the name of the profiled database in the PEM client tree control; navigate through the Management menu to the SQL Profiler pull-aside menu, and select Open trace... . You can also use the SQL Profiler toolbar menu to open a trace; select the Open trace... option. The Open trace... dialog opens.



Highlight an entry in the trace list and click Open to open the selected trace. The selected trace opens in the SQL Profiler tab.

Filtering a Trace

A filter is a named set of (one or more) rules, each of which can hide events from the trace view. When you apply a filter to a trace, the hidden events are not removed from the trace, but are merely excluded from the display. Click the Filter icon to open the Trace Filter dialog and create a rule (or set of rules) that define a filter. Each rule will screen the events within the current trace based on the identity of the role that invoked the event, or the query type invoked during the event.



To open an existing filter, select the Open button; to define a new filter, click the Add (+) button to add a row to the table displayed on the General tab

and provide rule details:

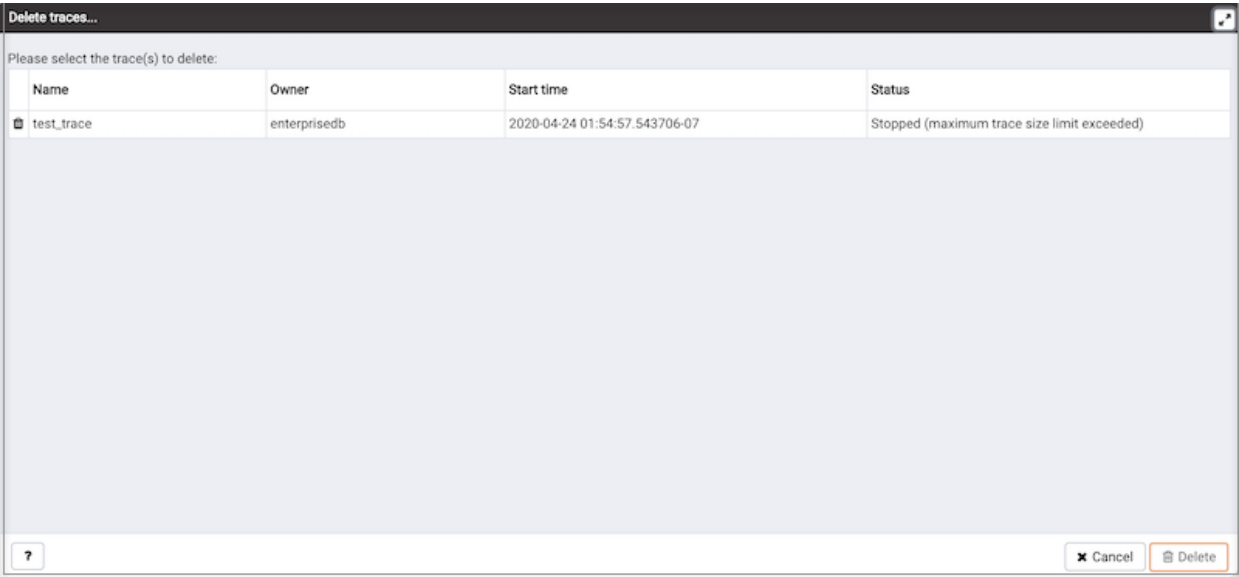
- Use the **Type** drop-down listbox to specify the trace field that the filter rule will apply to.
- Use the **Condition** drop-down listbox to specify the type of operator that SQL Profiler will apply to the Value when it filters the trace:
  - Select **Matches** to filter events that contain the specified Value.
  - Select **Does not match** to filter events that do not contain the specified Value.
  - Select **Is equal to** to filter events that contain an exact match to the string specified in the Value field.
  - Select **Is not equal to** to filter events that do not contain an exact match to the string specified in the Value field.
  - Select **Starts with** to filter events that begin with the string specified in the Value field.
  - Select **Does not start with** to filter events that do not begin with the string specified in the Value field.
  - Select **Less than** to filter events that have a numeric value less than the number specified in the Value field.
  - Select **Greater than** to filter events that have a numeric value greater than the number specified in the Value field.
  - Select **Less than or equal to** to filter events that have a numeric value less than or equal to the number specified in the Value field.
  - Select **Greater than or equal to** to filter events that have a numeric value greater than or equal to the number specified in the Value field.
- Use the **Value** field to specify the string, number or regular expression that SQL Profiler will search for.

When you've finished defining a rule, click the Add (+) button to add another rule to the filter. To delete a rule from a filter, highlight the rule and click the **Delete** icon.

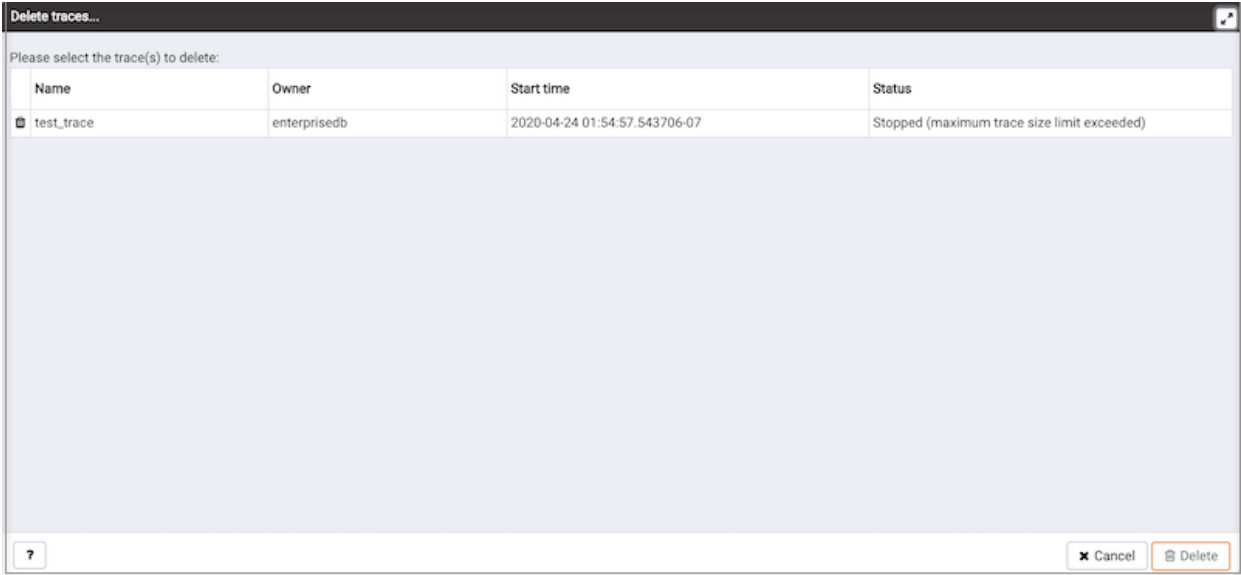
Click the **Save** button to save the filter definition to a file without applying the filter; to apply the filter, click **OK**. Select **Cancel** to exit the Trace Filter dialog and discard any changes to the filter.

Deleting a Trace

To delete a trace, highlight the name of the profiled database in the PEM client tree control; navigate through the **Management** menu to the **SQL Profiler** pull-aside menu, and select **Delete trace(s)...**. You can also use the SQL Profiler toolbar menu to delete a trace; select the **Delete trace(s)...** option. The **Delete traces...** dialog opens.



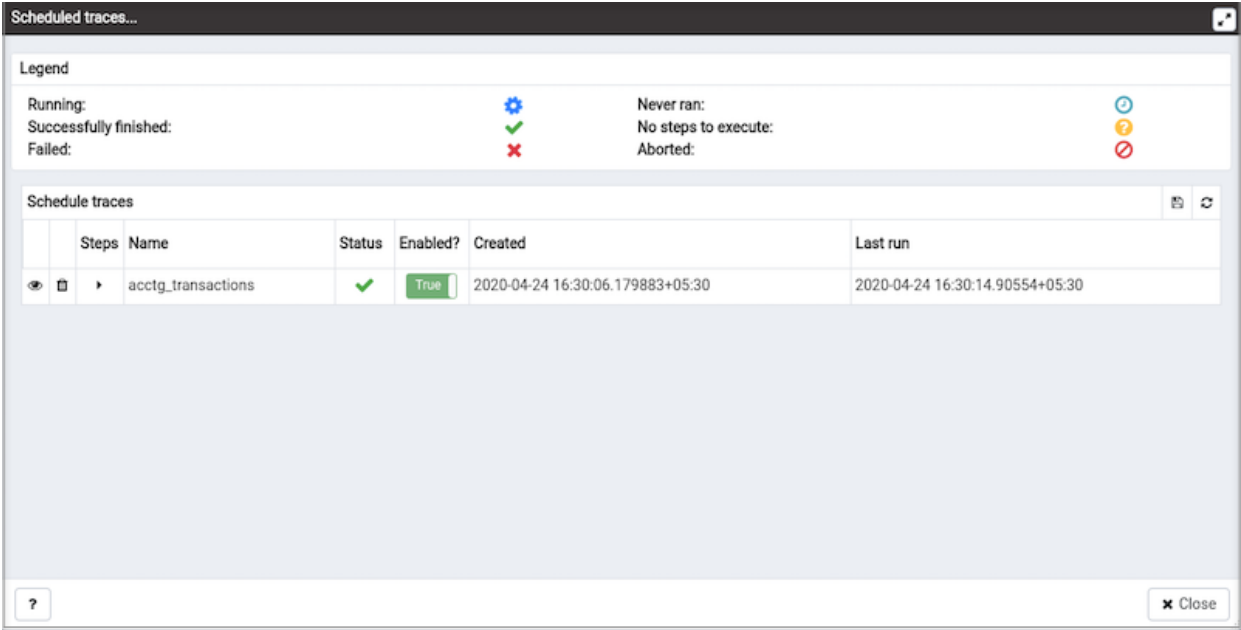
Click the icon to the left of a trace name to mark one or more traces for deletion and click **Delete**.



The PEM client will acknowledge that the selected traces have been deleted.

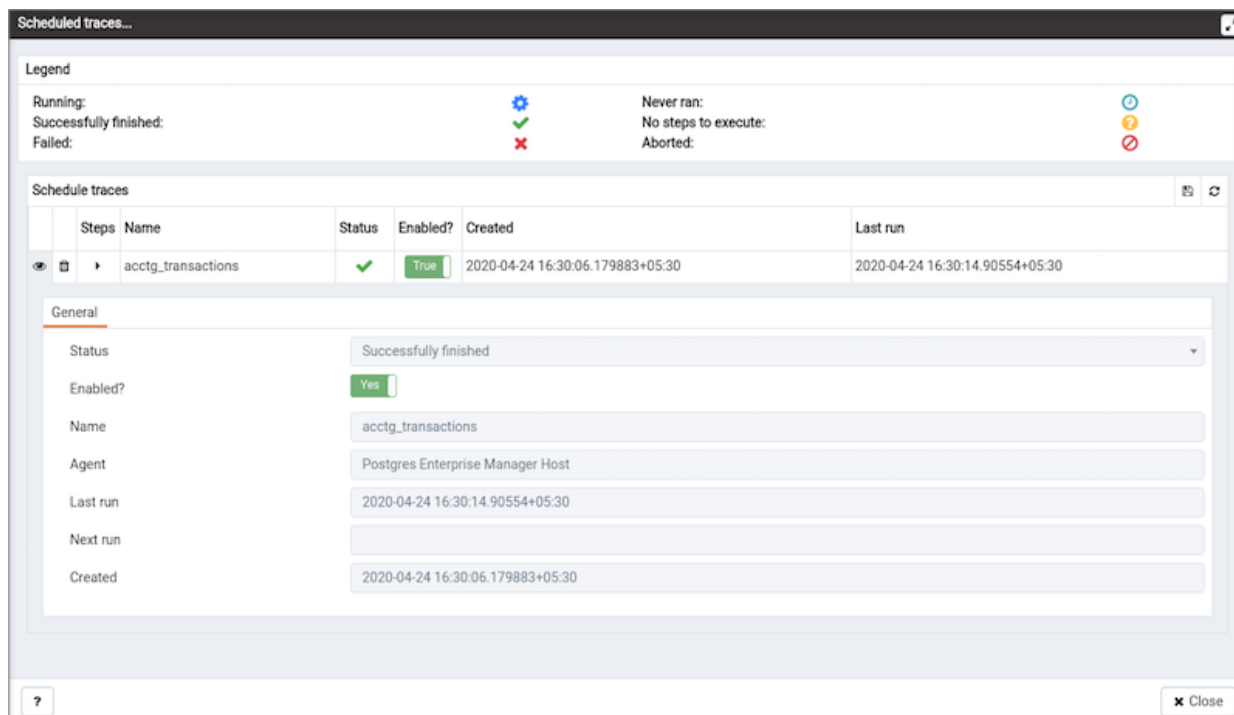
Viewing Scheduled Traces

To view a list of scheduled traces, highlight the name of the profiled database in the PEM client tree control; navigate through the **Management** menu to the **SQL Profiler** pull-aside menu, and select **Scheduled traces...**. You can also use the SQL Profiler toolbar menu to the list; select the **Scheduled traces...** option. The **Scheduled traces...** dialog opens.



The **Scheduled traces...** dialog displays a list of the traces that are awaiting execution. Click the edit button to the left of a trace name to access detailed information about the trace:





The **General** tab displays detailed information about the scheduled trace:

- The **Status** field lists the status of the current trace.
- The **Enabled?** switch displays **Yes** if the trace is enabled; **No** if the trace is disabled.
- The **Name** field displays the name of the trace.
- The **Agent** field displays the name of the agent responsible for executing the trace.
- The **Last run** field displays the date and time of the last execution of the trace.
- The **Next run** field displays the date and time of the next scheduled execution of the trace.
- The **Created** field displays the date and time that the trace was defined.

### 34.7.4 Using Index Advisor

Index Advisor helps you determine the application tables (and columns) on which you should create common B-tree type indexes. This can reduce the execution cost of queries you expect to use on your tables. Index Advisor comes pre-installed with EDB Postgres (R) Advanced Server. Index Advisor works with Advanced Server's query planner by creating "hypothetical indexes" for the query planner to use to calculate execution costs if such indexes were available.

Before using Index Advisor, you must:

1. Modify the postgresql.conf file on each Advanced Server host, adding the index\_advisor library to the shared\_preload\_libraries parameter.
2. Install the Index Advisor contrib module. To install the module, use the psql client or PEM Query tool to connect to the database, and invoke the following command:

```
<complete_path>/share/contrib/index_advisor.sql
```

3. Restart the server for your changes to take effect.

After installing Index Advisor, you can select one or more rows from within a trace, and select the Index Advisor icon to access Index Advisor functionality. For detailed installation and usage information about Index Advisor, please see the EDB Postgres Advanced Server Guide, available from the EnterpriseDB website at:

<http://www.enterprisedb.com>

Note

It is recommended that you disable the index advisor while using the pg\_dump functionality.

34.7.5 The SQL Profiler Tab

Toolbar Options

Toolbar options on the SQL Profiler tab allow you to define new traces, start or stop existing traces, open and search through previous traces, and filter trace results.



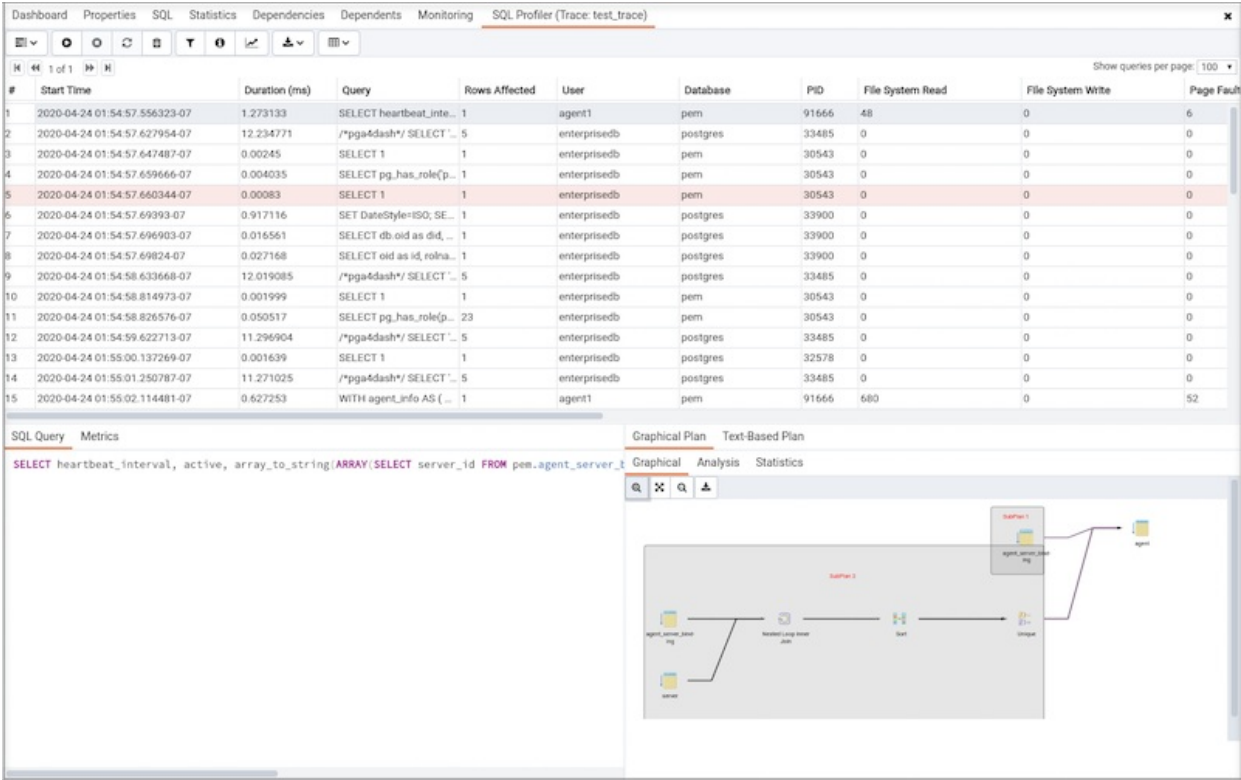
Use the following options to manage your SQL Profiler traces:

| Option         | Action  | Shortcut      |
|----------------|---|---------------|
| Menu           | Use options accessed through the drop_down menu icon to manage SQL Profiler traces.   | Accesskey + O |
| Start Trace    | Select the <b>Start Trace</b> icon to start a new trace, using the attributes (user names, database names, comments, etc) that were defined for the trace currently displayed in the SQL Profiler dialog. | Accesskey + S |
| Stop Trace     | Select the <b>Stop Trace</b> icon to stop an executing trace.   | Accesskey + Q |
| Refresh Trace  | Select the <b>Refresh Trace</b> icon to update the display to include any recent changes to an active trace.  | Accesskey + R |
| Clear Trace    | Select the <b>Clear Trace</b> icon to delete the trace and close the SQL Profiler window.   | Accesskey + C |
| Filter         | Select the <b>Filter</b> icon to define a new filter, or apply an existing filter to the trace.   | Accesskey + T |
| Information    | Select the information icon to view the properties of the trace displayed in the SQL Profiler window.   | Accesskey + P |
| Index Advisor  | Select the <b>Index Advisor</b> icon to open the <a href="#">PEM Index Advisor</a> .   Accesskey + I  |               |
| Download Trace | Use options accessed through the <b>Download Trace</b> menu to download a CSV file that contains the trace events shown on the current page or the complete set of trace data.                            | Accesskey + X |
| Column Picker  | Click the <b>Column Picker</b> icon to choose the columns to be displayed in below table.   | Accesskey + W |

Viewing Trace Data

The SQL Profiler tab is divided into three panes:

- The top of the tabbed browser window (the trace data pane) displays a list of the SQL commands executed during the trace.
- The lower-left panel displays the SQL query that was executed, or the metrics gathered during the execution.
- The lower-right panel displays the query execution plan; you can view the execution plan in a Text-based form, or as a Graphical Plan.



The Trace Data Pane

The Trace Data pane displays the SQL commands executed during the trace. By default, the commands are displayed in the order that the command was executed.

Double-click a column heading to sort the trace by the column values; double-click the column heading a second time to toggle the data in the column to be in ascending or descending order.

Use the drop-down listbox next to the `Show queries per page` label to specify the number of events that SQL Profiler should display in the pane. Select from 20, 50, 100, 200, 500, 1000, or 2000. The default is 500.

If the number of events in the trace exceeds the count of events per page, use the page selector controls (located in the top left corner of the table) to navigate through pages of the trace.

To include or exclude events from the currently displayed trace, select the Filter icon from the SQL Profiler toolbar. The Trace Filter dialog will allow you to define and apply a filter that will screen the displayed trace.

The Query/Metrics Pane

The Query/Metrics pane is located in the lower-left corner of the SQL Profiler window. The tabs provide detailed information about the currently selected query:

- The `SQL Query` tab displays the text of the query that is currently highlighted in Trace Data pane.
- The `Metrics` tab displays detailed statistical information about the execution of the query. The table below describes the metrics that are displayed in the Metrics dialog; the percentages listed describe the percentage of the total quantity of the parameter that is attributed to the selected SQL command:

| Property      | Description  |
|---------------|--|
| Executed (#)  | The number of times that the selected SQL command executed.  |
| Execution (%) | The percentage of the execution count that the SQL command represents. For example if the trace profiles 4 SQL commands, each command will represent 25% of the trace execution %. |
| Duration (%)  | The percentage of the total trace time consumed by the highlighted SQL Command.  |

| Property                     | Description   |
|------------------------------|---|
| Rows updated (%)             | The percentage of the rows updated during the trace that were updated by the selected SQL command.                |
| Page faults (%)              | The percentage of the page faults that occur during the trace that can be attributed to the selected SQL command. |
| Page reclaims (%)            | The percentage of the pages reclaimed during the trace that can be attributed to the selected SQL command.        |
| Swaps (%)                    | The percentage of swaps that occur during the trace that can be attributed to the selected SQL command.           |
| File system in (%)           | The percentage of bytes written to disk during the trace that can be attributed to the selected SQL command.      |
| File system out (%)          | The percentage of bytes read from disk during the trace that can be attributed to the selected SQL command.       |
| Signals received (%)         | Currently unused.   |
| Messages received            | (%)Currently unused.  |
| Messages sent (%)            | Currently unused.   |
| Voluntary context switches   | (%)Currently unused.  |
| Involuntary context switches | (%)Currently unused.  |
| Shared blocks read (%)       | The percentage of the shared blocks read by the highlighted SQL command.  |
| Shared blocks written (%)    | The percentage of the shared blocks written by the highlighted SQL command.                                       |
| Shared blocks hit (%)        | The percentage of the shared blocks hit by the highlighted SQL command.   |
| Local blocks read (%)        | The percentage of local blocks read by the highlighted SQL command.   |
| Local blocks written (%)     | The percentage of local blocks written by the highlighted SQL command.  |
| Local blocks hit (%)         | The percentage of local blocks hit by the highlighted SQL commands.   |
| Temporary blocks read (%)    | The percentage of the temporary blocks read by the highlighted SQL commands.                                      |
| Temporary blocks written (%) | The percentage of the temporary blocks written by the highlighted SQL commands.                                   |

## The Explain Pane

The Graphical or Text-based explain pane displays one of two representations of the query execution plan for the selected query.

- Select the Text-based Plan tab to display the execution plan for the currently highlighted event in text format:
- Select the Graphical-based Plan tab to display a graphical interpretation of the execution plan of the highlighted query. For more information about interpreting the graphical query plan, see [Interpreting the Graphical Query Plan](#).

## 34.8 Developer Tools

The PEM client features powerful developer tools that you can use to execute and analyze complex SQL commands, manage data, and debug PL/SQL code.

### 34.8.1 PL Debugger

agent\_down\_status(p\_agent\_id integer, p\_start\_datetime timestamp with time zone, p\_end\_datetime timestamp with time zone, o\_down\_datetime timestamp with time zone, o\_up\_datetime timestamp with time zone)

2 DECLARE

3 v\_alert\_id integer;

4 v\_curr\_rec record;

5 v\_prev\_state pem.alert\_state := NULL;

6 BEGIN

7 SELECT id INTO v\_alert\_id FROM pem.alert WHERE agent\_id = p\_agent\_id and template\_id = (SELECT id FROM pem.alert\_template WHERE display\_name = 'Agent Down');

8 o\_down\_datetime := NULL;

9 o\_up\_datetime := NULL;

10

11 FOR v\_curr\_rec IN EXECUTE '

12 SELECT

13 state, generated as recorded\_time

14 FROM

15 pem.alert\_history

16 WHERE

17 alert\_id = \$1::integer AND generated >= \$2::timestampz AND generated <= \$3::timestampz

18 ORDER BY generated;' USING v\_alert\_id, p\_start\_datetime, p\_end\_datetime

Parameters

Local variables

Messages

Results

Stack

| Name             | Type                     | Value |
|------------------|--------------------------|-------|
| p_agent_id       | integer                  | 1     |
| p_start_datetime | timestamp with time zone | NULL  |
| p_end_datetime   | timestamp with time zone | NULL  |
| o_down_datetime  | timestamp with time zone | NULL  |
| o_up_datetime    | timestamp with time zone | NULL  |

The debugger may be used to debug PL/pgSQL functions in PostgreSQL, as well as EDB-SPL functions, stored procedures and packages in Advanced Server. The Debugger is available as an extension for your PostgreSQL installation, and is distributed as part of Advanced Server. You must have superuser privileges to use the debugger.

Before using the debugger, you must modify the `postgresql.conf` file, adding the server-side debugger components to the the value of the `shared_preload_libraries` parameter:

`shared_preload_libraries = 'slibdir/other_libraries/plugin_debugger'`

After modifying the `shared_preload_libraries` parameter, restart the server to apply the changes.

The debugger may be used for either in-context debugging or direct debugging of a target function or procedure. When you use the debugger for in-context debugging, you set a breakpoint at the first line of a program; when a session invokes the target, control is transferred to the debugger. When using direct debugging, the debugger prompts you for any parameters required by the target, and then allows you to step through the code.

In-context Debugging

To set a breakpoint at the first line of a program, right-click the name of the object you would like to debug, and select `Set breakpoint` from the `Debugging` sub-menu. The debugger window will open, waiting for another session to invoke the program.

agent\_down\_status(p\_agent\_id integer, p\_start\_datetime timestamp with time zone, p\_end\_datetime timestamp with time zone, o\_down\_datetime timestamp with time zone, o\_up\_datetime timestamp with time zone)

1

Waiting for another session to invoke the target...

Parameters

Local variables

Messages

Results

Stack

When another session invokes the target, the debugger will display the code, allowing you to add break points, or step through line-by-line. The other session is suspended until the debugging completes; then control is returned to the session.

agent\_down\_status(p\_agent\_id integer, p\_start\_datetime timestamp with time zone, p\_end\_datetime timestamp with time zone, o\_down\_datetime timestamp with time zone, o\_up\_datetime timestamp with time zone)

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

DECLARE

v\_alert\_id integer;

v\_curr\_rec record;

v\_prev\_state pem.alert\_state := NULL;

BEGIN

SELECT id INTO v\_alert\_id FROM pem.alert WHERE agent\_id = p\_agent\_id and template\_id = (SELECT id FROM pem.alert\_template WHERE display\_name = 'Agent

o\_down\_datetime := NULL;

o\_up\_datetime := NULL;

FOR v\_curr\_rec IN EXECUTE '

SELECT

state, generated as recorded\_time

FROM

pem.alert\_history

WHERE

alert\_id = \$1::integer AND generated >= \$2::timestampz AND generated <= \$3::timestampz

Parameters

Local variables

Messages

Results

Stack

| Name   | Value  | Line No. |
|--|--|----------|
| pem.agent_down_status(integer,timestamp with time zone,timestamp with time zone) | p_agent_id=1, p_start_datetime=, p_end_datetime= | 7        |

Direct Debugging

To use the debugger for direct debugging, right click on the name of the object that you wish to debug in the browser tree control and select **Debug** from the **Debugging** sub-menu. The debugger window will open, prompting you for any values required by the program:

Debugger

| Name             | Type                     | Null?                               | Expression?              | Value | Use Default?             | Default      |
|------------------|--------------------------|-------------------------------------|--------------------------|-------|--------------------------|--------------|
| p_agent_id       | integer                  | <input type="checkbox"/>            | <input type="checkbox"/> | 1     | <input type="checkbox"/> | <no default> |
| p_start_datetime | timestamp with time zone | <input checked="" type="checkbox"/> | <input type="checkbox"/> |       | <input type="checkbox"/> | <no default> |
| p_end_datetime   | timestamp with time zone | <input checked="" type="checkbox"/> | <input type="checkbox"/> |       | <input type="checkbox"/> | <no default> |

Clear All

✕

Cancel

Debug

Use the fields on the **Debugger** dialog to provide a value for each parameter:

- The **Name** field contains the formal parameter name.
- The **Type** field contains the parameter data type.
- Check the **Null?** checkbox to indicate that the parameter is a NULL value.
- Check the **Expression?** checkbox if the Value field contains an expression.
- Use the **Value** field to provide the parameter value that will be passed to the program. When entering parameter values, type the value into the appropriate cell on the grid, or, leave the cell empty to represent NULL, enter " (two single quotes) to represent an empty string, or to enter a literal string consisting of just two single quotes, enter ". PostgreSQL 8.4 and above supports variadic function parameters. These may be entered as a comma-delimited list of values, quoted and/or cast as required.
- Check the **Use default?** checkbox to indicate that the program should use the value in the Default Value field.
- The **Default Value** field contains the default value of the parameter.

Provide values required by the program, and click the **Debug** button to start stepping through the program. The values of the arguments provided here are saved. The values will be pre-filled next time the dialog opens. To clear the values, use the **Clear All** button.

agent\_down\_status(p\_agent\_id integer, p\_start\_datetime timestamp with time zone, p\_end\_datetime timestamp with time zone, o\_down\_datetime timestamp with time zone, o\_up\_datetime timestamp with time zone)

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

DECLARE

v\_alert\_id integer;

v\_curr\_rec record;

v\_prev\_state pem.alert\_state := NULL;

BEGIN

SELECT id INTO v\_alert\_id FROM pem.alert WHERE agent\_id = p\_agent\_id and template\_id = (SELECT id FROM pem.alert\_template WHERE display\_name = 'Agent

o\_down\_datetime := NULL;

o\_up\_datetime := NULL;

FOR v\_curr\_rec IN EXECUTE '

SELECT

state, generated as recorded\_time

FROM

pem.alert\_history

WHERE

alert\_id = \$1::integer AND generated >= \$2::timestampz AND generated <= \$3::timestampz

Parameters

Local variables

Messages

Results

Stack

| Name   | Value  | Line No. |
|--|--|----------|
| pem.agent_down_status(integer,timestamp with time zone,timestamp with time zone) | p_agent_id=1, p_start_datetime=, p_end_datetime= | 7        |

Using the Debugger

The main debugger window consists of two panels and a context-sensitive toolbar. Use toolbar icons to manage breakpoints and step into or through code; hover over an icon for a tooltip that identifies the option associated with the icon. The toolbar options are:



| Option                | Action   |
|-----------------------|--|
| Step into             | Click the <b>Step into</b> icon to execute the currently highlighted line of code.   |
| Step over             | Click the <b>Step over</b> icon to execute a line of code, stepping over any sub-functions invoked by the code.The sub-function executes, but is not debugged unless it contains a breakpoint. |
| Continue/Start        | Click the <b>Continue/Start</b> icon to execute the highlighted code, and continue until the programencounters a breakpoint or completes.  |
| Toggle breakpoint     | Use the <b>Toggle breakpoint</b> icon to enable or disable a breakpoint (without removing the breakpoint).   |
| Clear all breakpoints | Click the <b>Clear all breakpoints</b> icon to remove all breakpoints from the program.  |
| Stop                  | Click the <b>Stop</b> icon to halt the execution of a program.   |

The top panel of the debugger window displays the program body; click in the grey margin next to a line number to add a breakpoint. The highlighted line in the top panel is the line that is about to execute.

agent\_down\_status(p\_agent\_id integer, p\_start\_datetime timestamp with time zone, p\_end\_datetime timestamp with time zone, o\_down\_datetime timestamp with time zone, o\_up\_datetime timestamp with time zone)

2 DECLARE

3 v\_alert\_id integer;

4 v\_curr\_rec record;

5 v\_prev\_state pem.alert\_state := NULL;

6 BEGIN

7 SELECT id INTO v\_alert\_id FROM pem.alert WHERE agent\_id = p\_agent\_id and template\_id = (SELECT id FROM pem.alert\_template WHERE display\_name = 'Agent Down');

8 o\_down\_datetime := NULL;

9 o\_up\_datetime := NULL;

10

11 FOR v\_curr\_rec IN EXECUTE '

12 SELECT

13 state, generated as recorded\_time

14 FROM

15 pem.alert\_history

16 WHERE

17 alert\_id = \$1::integer AND generated >= \$2::timestampz AND generated <= \$3::timestampz

18 ORDER BY generated;' USING v\_alert\_id, p\_start\_datetime, p\_end\_datetime

Parameters

Local variables

Messages

Results

Stack

| Name             | Type                     | Value |
|------------------|--------------------------|-------|
| p_agent_id       | integer                  | 1     |
| p_start_datetime | timestamp with time zone | NULL  |
| p_end_datetime   | timestamp with time zone | NULL  |
| o_down_datetime  | timestamp with time zone | NULL  |
| o_up_datetime    | timestamp with time zone | NULL  |

The lower panel of the debugger window provides a set of tabs that allow you to review information about the program:

- The **Parameters** tab displays the value of each parameter.
- The **Local variables** tab displays the current value of the program variables.
- The **Messages** tab displays any messages returned by the server (errors, warnings and informational messages).
- The **Results** tab displays the server message when the program completes.
- The **Stack** tab displays the list of functions that have been invoked, but which have not yet completed.

As you step through a program, the **Local variables** tab displays the current value of each variable:

Parameters

Local variables

Messages

Results

Stack

| Name         | Type            | Value |
|--------------|-----------------|-------|
| v_alert_id   | integer         | NULL  |
| v_prev_state | pem.alert_state | NULL  |

When you step into a subroutine, the **Stack** tab displays the call stack, including the name of each caller, the parameter values for each caller (if any), and the line number within each caller:

| Parameters   | Local variables | Messages   | Results  | <u>Stack</u> |
|--|-----------------|--|----------|--------------|
| Name   |                 | Value  | Line No. |              |
| pem.agent_down_status(integer,timestamp with time zone,timestamp with time zone) |                 | p_agent_id=1, p_start_datetime=, p_end_datetime= | 7        |              |
|  |                 |  |          |              |

Select a caller to change focus to that stack frame and display the state of the caller in the upper panel.

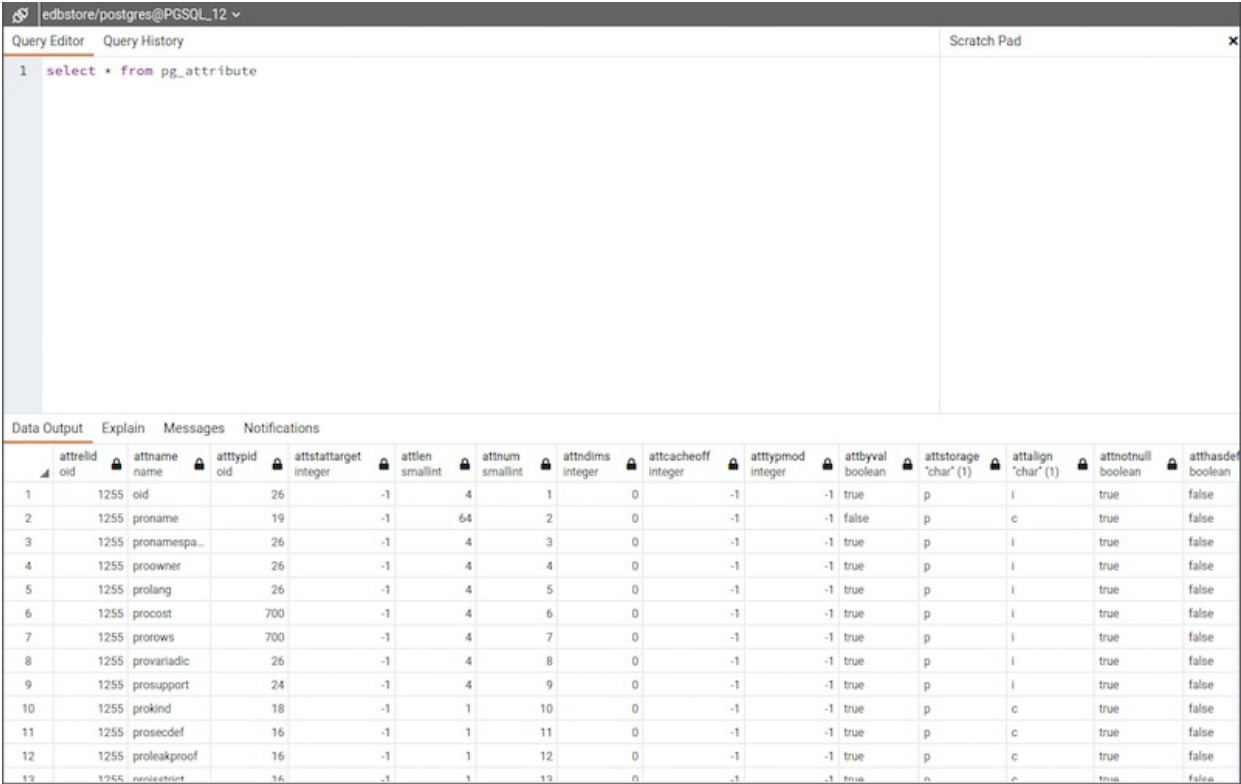
When the program completes, the **Results** tab displays the message returned by the server. If the program encounters an error, the **Messages** tab displays details:



If `ENABLE_DEBUGGER` configuration option is set to `False` then debugger is disabled.

The Query tool is a powerful, feature-rich environment that allows you to execute arbitrary SQL commands and review the result set. You can access the Query tool via the **Query tool** menu option on the **Tools** menu, or through the context menu of select nodes of the Browser tree control. The Query tool allows you to:

- Issue ad-hoc SQL queries.
- Execute arbitrary SQL commands.
- Edit the result set of a SELECT query if it is [updatable](#).
- Displays current connection and transaction status as configured by the user.
- Save the data displayed in the output panel to a CSV file.
- Review the execution plan of a SQL statement in either a text, a graphical format or a table format (similar to <https://explain.depesz.com>).
- View analytical information about a SQL statement.



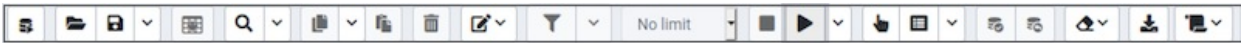
You can open multiple copies of the Query tool in individual tabs simultaneously. To close a copy of the Query tool, click the **X** in the upper-right hand corner of the tab bar.

The Query tool features two panels:

- The upper panel displays the **SQL Editor**. You can use the panel to enter, edit, or execute a query. It also shows the **History** tab which can be used to view the queries that have been executed in the session, and a **Scratch Pad** which can be used to hold text snippets during editing. If the Scratch Pad is closed, it can be re-opened (or additional ones opened) by right-clicking in the SQL Editor and other panels and adding a new panel.
- The lower panel displays the **Data Output** panel. The tabbed panel displays the result set returned by a query, information about a query's execution plan, server messages related to the query's execution and any asynchronous notifications received from the server.

The Query tool Toolbar

The **Query tool** toolbar uses context-sensitive icons that provide shortcuts to frequently performed tasks. If an icon is highlighted, the option is enabled; if the icon is grayed-out, the task is disabled. Please note that disabled icons may support functionality accessed via the **data editor**.



Hover over an icon to display a tooltip that describes the icon's functionality:

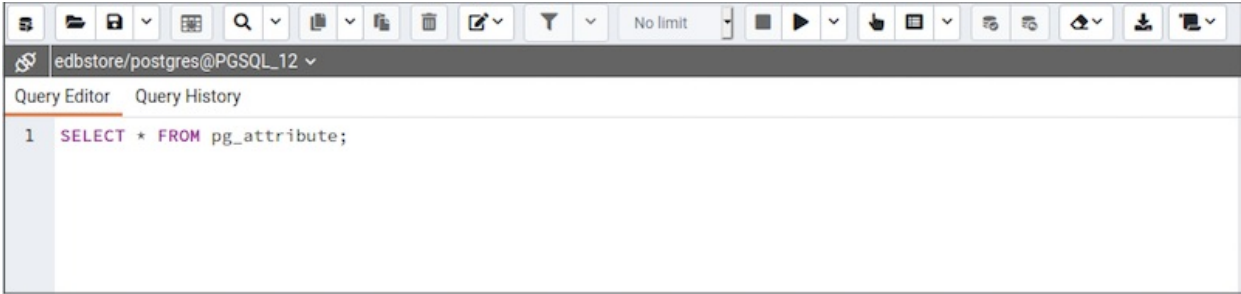
| Icon      | Behavior   | Shortc<br>ut         |
|-----------|--|----------------------|
| Open File | Click the <b>Open File</b> icon to display a previously saved query in the SQL Editor.   | Access<br>key +<br>O |
| Save      | Click the <b>Save</b> icon to perform a quick-save of a previously saved query, or to access the <b>Save</b> menu:<br>- Select <b>Save</b> to save the selected content of the SQL Editor panel in a file.<br>- Select <b>Save As</b> to open a new browser dialog and specify a new location to which to save the selected content of the SQL Editor panel. | Access<br>key +<br>S |

| Icon                   | Behavior  | Shortcut  |
|------------------------|---|---|
|                        |   | Cmd+F   |
| <b>Find</b>            | <p>Use the <b>Find</b> menu to search, replace, or navigate the code displayed in the SQL Editor:</p> <ul style="list-style-type: none"> <li>- Select <b>Find</b> to provide a search target, and search the SQL Editor contents.</li> <li>- Select <b>Find next</b> to locate the next occurrence of the search target.</li> <li>- Select <b>Find previous</b> to move to the last occurrence of the search target.</li> <li>- Select <b>Persistent find</b> to identify all occurrences of the search target within the editor.</li> <li>- Select <b>Replace</b> to locate and replace (with prompting) individual occurrences of the target.</li> <li>- Select <b>Replace all</b> to locate and replace all occurrences of the target within the editor.</li> <li>- Select <b>Jump</b> to navigate to the next occurrence of the search target.</li> </ul> | <p>Cmd+G</p> <p>Cmd+Shift+G</p> <p>Cmd+Shift+F</p> <p>Alt+G</p>                 |
| <b>Copy</b>            | Click the <b>Copy</b> icon to copy the content that is currently highlighted in the Data Output panel. when in View/Edit data mode.   | Access key + C  |
| <b>Paste</b>           | Click the <b>Paste</b> icon to paste a previously row into a new row when in View/Edit data mode.   | Access key + P  |
| <b>Delete</b>          | Click the <b>Delete</b> icon to mark the selected rows for delete when in View/Edit data mode.  | Access key + D  |
| <b>Edit</b>            | <p>Use options on the <b>Edit</b> menu to access text editing tools; the options operate on the text displayed in the SQL Editor panel when in Query tool mode:</p> <ul style="list-style-type: none"> <li>- Select <b>Indent Selection</b> to indent the currently selected text.</li> <li>- Select <b>Unindent Selection</b> to remove indentation from the currently selected text.</li> <li>- Select <b>Inline Comment Selection</b> to enclose any lines that contain the selection in SQL style comment notation.</li> <li>- Select <b>Inline Uncomment Selection</b> to remove SQL style comment notation from the selected line.</li> <li>- Select <b>Block Comment</b> to enclose all lines that contain the selection in C style comment notation. This option acts as a toggle.</li> </ul>   | <p>Tab</p> <p>Shift+Tab</p> <p>Cmd+/<br/>Cmd+.</p> <p>Shift+Cmd+/<br/>Cmd+.</p> |
| <b>Filter</b>          | <p>Click the <b>Filter</b> icon to set filtering and sorting criteria for the data when in View/Edit data mode. Click the down arrow to access other filtering and sorting options:</p> <ul style="list-style-type: none"> <li>- Click <b>Sort/Filter</b> to open the sorting and filtering dialogue.</li> <li>- Click <b>Filter by Selection</b> to show only the rows containing the values in the selected cells.</li> <li>- Click <b>Exclude by Selection</b> to show only the rows that do not contain the values in the selected cells.</li> <li>- Click <b>Remove Sort/Filter</b> to remove any previously selected sort or filtering options.</li> </ul>  | Access key + F  |
| <b>Limit Selector</b>  | Select a value in the <b>Limit Selector</b> to limit the size of the dataset to a number of rows.   | Access key + R  |
| <b>Stop</b>            | Click the <b>Stop</b> icon to cancel the execution of the currently running query.  | Access key + Q  |
| <b>Execute/Refresh</b> | <p>Click the <b>Execute/Refresh</b> icon to either execute or refresh the query highlighted in the SQL editor panel. Click the down arrow to access other execution options:</p> <ul style="list-style-type: none"> <li>- Add a check next to <b>Auto-Rollback</b> to instruct the server to automatically roll back a transaction if an error occurs during the transaction.</li> <li>- Add a check next to <b>Auto-Commit</b> to instruct the server to automatically commit each transaction. Any changes made by the transaction will be visible to others, and durable in the event of a crash.</li> </ul>   | F5  |
| <b>Explain</b>         | <p>- Click the <b>Explain</b> icon to view an explanation plan for the current query. The result of the EXPLAIN is displayed graphically on the <b>Explain</b> tab of the output panel, and in text form on the <b>Data Output</b> tab.</p>   | F7  |

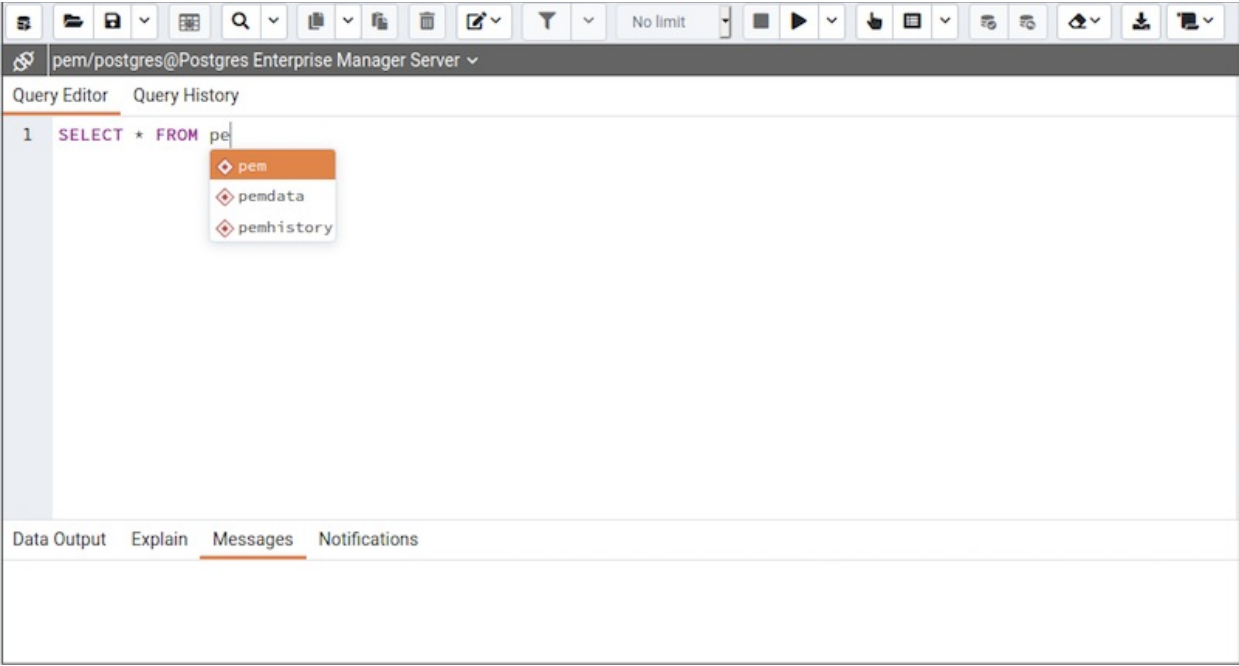
| Icon                   | Behavior  | Shortcut       |
|------------------------|---|----------------|
|                        | Click the <b>Explain analyze</b> icon to invoke an EXPLAIN ANALYZE command on the current query.  |                |
| <b>Explain analyze</b> | Navigate through the <b>Explain Options</b> menu to select options for the EXPLAIN command: <ul style="list-style-type: none"><li>- Select <b>Verbose</b> to display additional information regarding the query plan.</li><li>- Select <b>Costs</b> to include information on the estimated startup and total cost of each plan node, as well as the estimated number of rows and the estimated width of each row.</li><li>- Select <b>Buffers</b> to include information on buffer usage.</li><li>- Select <b>Timing</b> to include information about the startup time and the amount of time spent in each node of the query.</li></ul> | Shift+F7       |
| <b>Commit</b>          | Click the <b>Commit</b> icon to commit the transaction.   | Shift+CTRL+M   |
| <b>Rollback</b>        | Click the <b>Rollback</b> icon to rollback the transaction.   | Shift+CTRL+R   |
| <b>Clear</b>           | Use options on the <b>Clear</b> drop-down menu to erase display contents: <ul style="list-style-type: none"><li>- Select <b>Clear Query Window</b> to erase the content of the SQL Editor panel.</li><li>- Select <b>Clear History</b> to erase the content of the <b>History</b> tab.</li></ul>  | Access key + L |
| <b>Download as CSV</b> | Click the <b>Download as CSV</b> icon to download the result set of the current query to a comma-separated list. You can specify the CSV settings through <b>Preferences -&gt; SQL Editor -&gt; CSV output</b> dialogue.  | F8             |
| <b>Macros</b>          | Click the <b>Macros</b> icon to manage the macros. You can create, edit or clear the macros through <i>Manage Macros</i> option.  |                |

The SQL Editor Panel

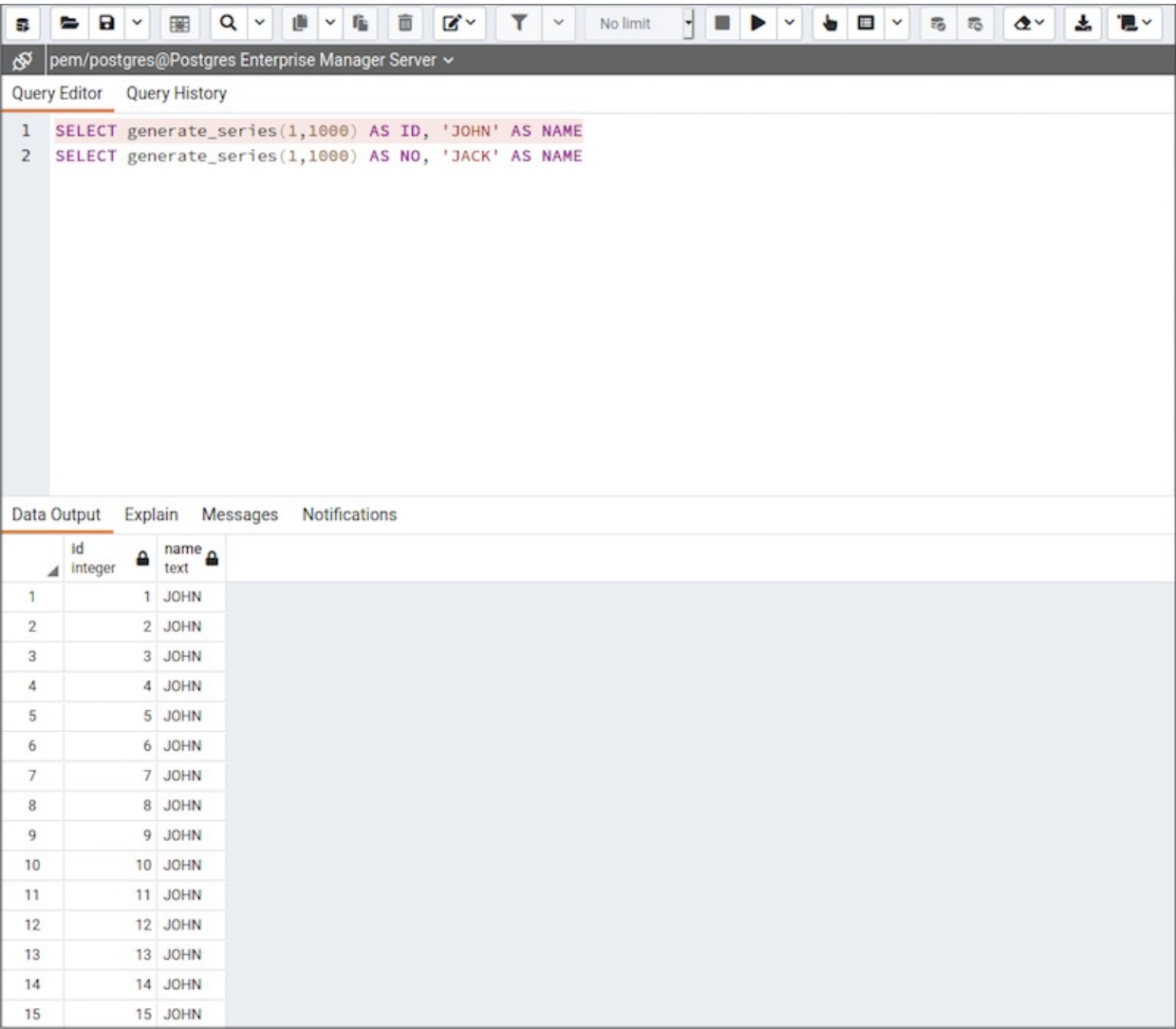
The **SQL editor** panel is a workspace where you can manually provide a query, copy a query from another source, or read a query from a file. The SQL editor features syntax coloring and autocompletion.



To use autocomplete, begin typing your query; when you would like the Query editor to suggest object names or commands that might be next in your query, press the Control+Space key combination. For example, type "SELECT \* FROM" (without quotes, but with a trailing space), and then press the Control+Space key combination to select from a popup menu of autocomplete options.

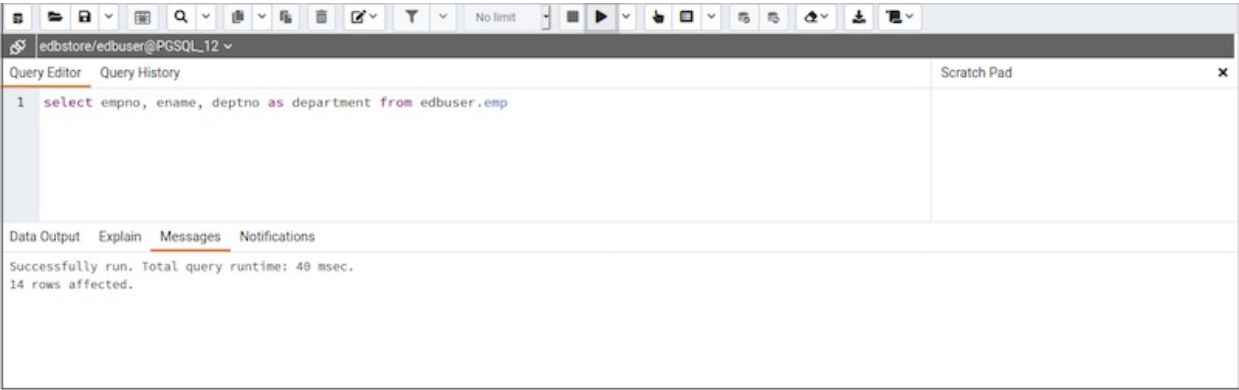


After entering a query, select the **Execute/Refresh** icon from the toolbar. The complete contents of the SQL editor panel will be sent to the database server for execution. To execute only a section of the code that is displayed in the SQL editor, highlight the text that you want the server to execute, and click the **Execute/Refresh** icon.



The message returned by the server when a command executes is displayed on the **Messages** tab. If the command is successful, the **Messages** tab

displays execution details.



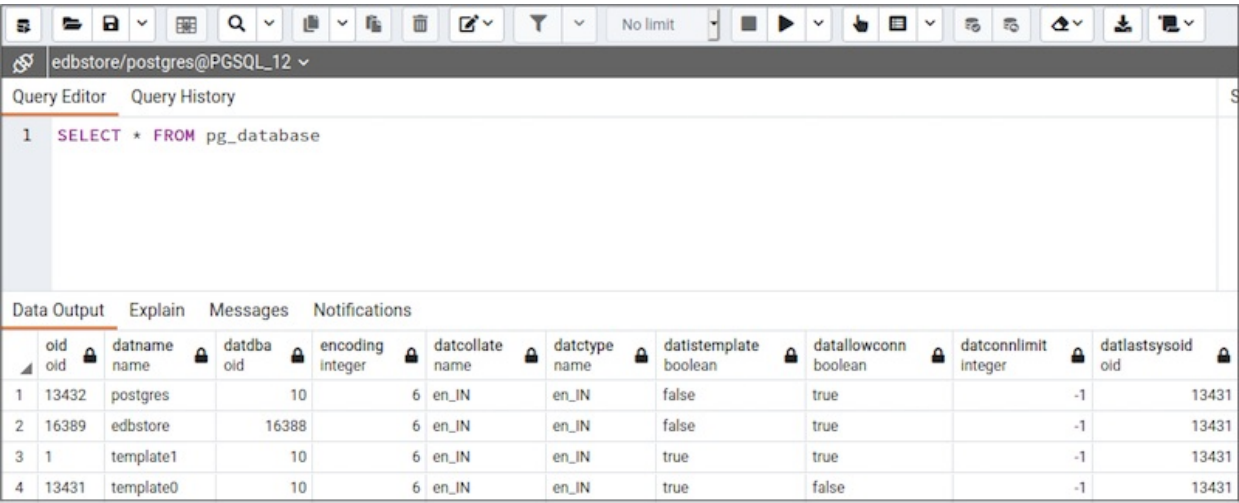
Options on the **Edit** menu offer functionality that helps with code formatting and commenting:

- The auto-indent feature will automatically indent text to the same depth as the previous line when you press the Return key.
- Block indent text by selecting two or more lines and pressing the Tab key.
- Implement or remove SQL style or toggle C style comment notation within your code.

You can also **drag and drop** certain objects from the treeview which can save time in typing long object names. Text containing the object name will be fully qualified with schema. Double quotes will be added if required. For functions and procedures, the function name along with parameter names will be pasted in the Query tool.

The Data Output Panel

The **Data Output** panel displays data and statistics generated by the most recently executed query.



The **Data Output** tab displays the result set of the query in a table format. You can:

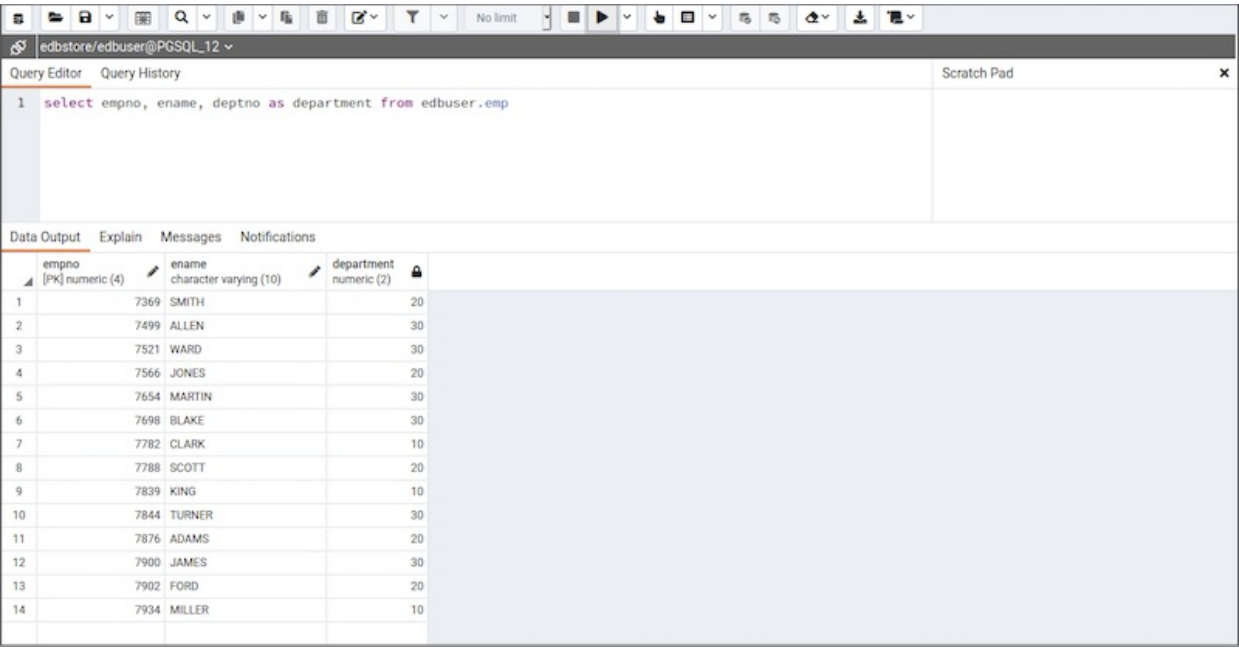
- Select and copy from the displayed result set.
- Use the **Execute/Refresh** options to retrieve query execution information and set query execution options.
- Use the **Save results to file** icon to save the content of the **Data Output** tab as a comma-delimited file.
- Edit the data in the result set of a SELECT query if it is updatable.

A result set is updatable if:

- All columns are either selected directly from a single table, or are not table columns at all (e.g. concatenation of 2 columns). Only columns that are selected directly from the table are editable, other columns are read-only.
- All the primary key columns or OIDs of the table are selected in the result set.

Any columns that are renamed or selected more than once are also read-only.

Editable and read-only columns are identified using pencil and lock icons (respectively) in the column headers.



The psycopg2 driver version should be equal to or above 2.8 for updatable query result sets to work.

An updatable result set is identical to the [Data Grid](#) in View/Edit Data mode, and can be modified in the same way.

If Auto-commit is off, the data changes are made as part of the ongoing transaction, if no transaction is ongoing a new one is initiated. The data changes are not committed to the database unless the transaction is committed.

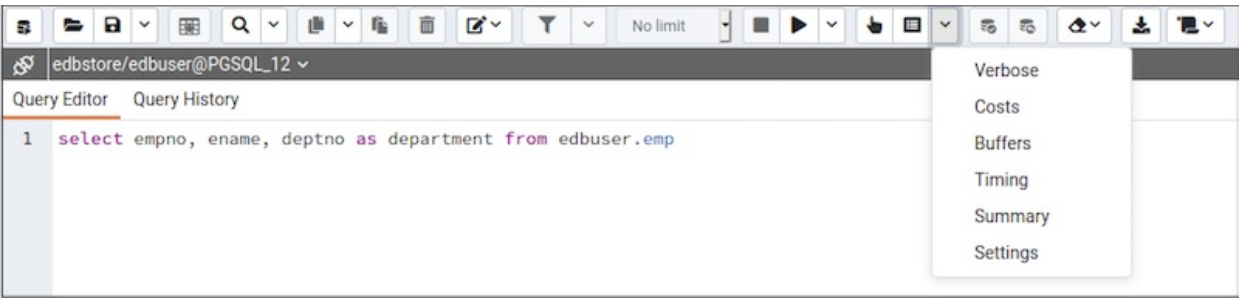
If any errors occur during saving (for example, trying to save NULL into a column with NOT NULL constraint) the data changes are rolled back to an automatically created SAVEPOINT to ensure any previously executed queries in the ongoing transaction are not rolled back.

All rowsets from previous queries or commands that are displayed in the [Data Output](#) panel will be discarded when you invoke another query; open another query tool browser tab to keep your previous results available.

Explain Panel

To generate the [Explain](#) or [Explain Analyze](#) plan of a query, click on [Explain](#) or [Explain Analyze](#) button in the toolbar.

More options related to [Explain](#) and [Explain Analyze](#) can be selected from the drop down on the right side of [Explain Analyze](#) button in the toolbar.



Please note that PEM generates the [Explain \[Analyze\]](#) plan in JSON format.



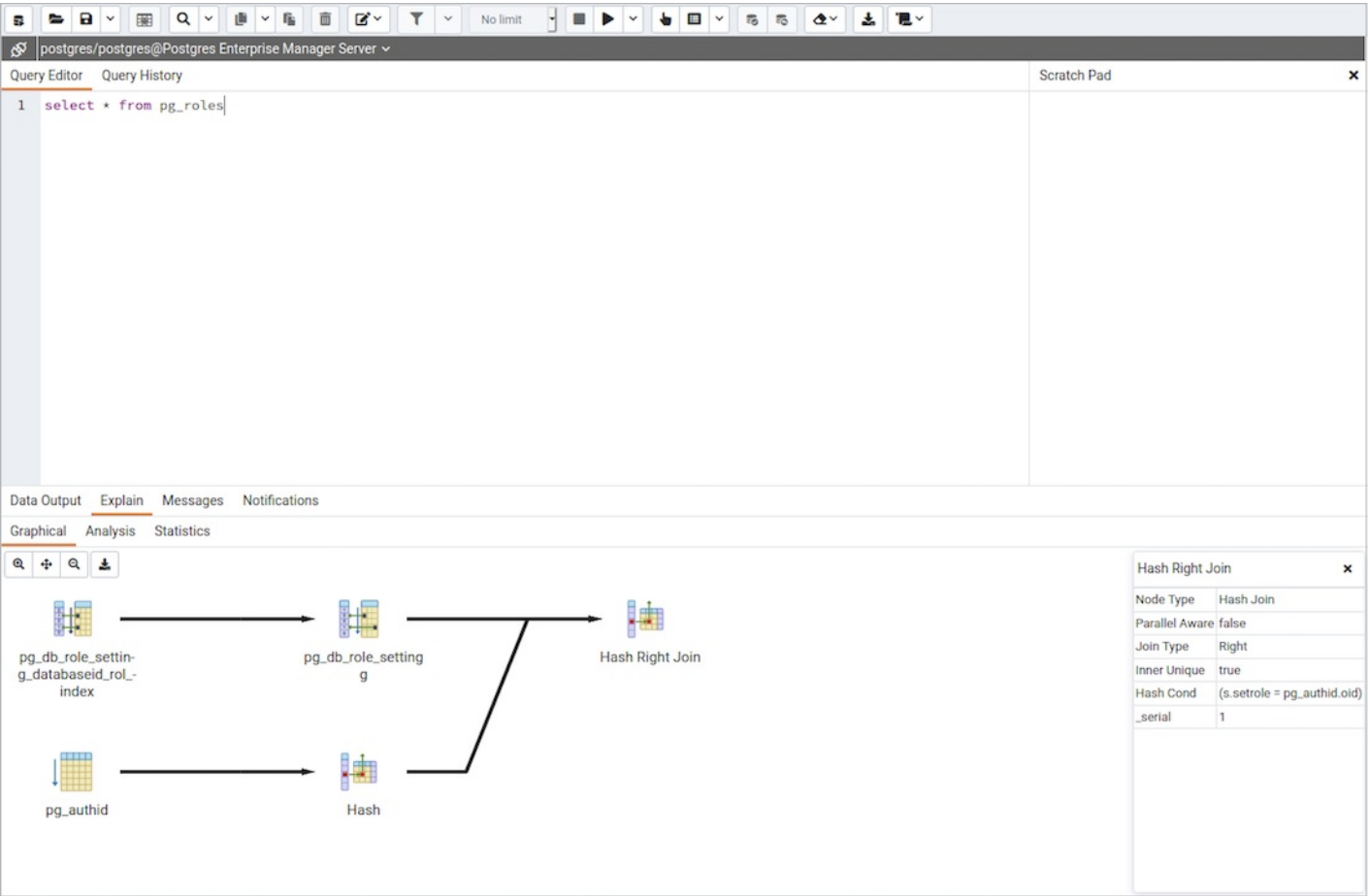
On successful generation of **Explain** plan, it will create three tabs/panels under the Explain panel.

- Graphical

Please note that **EXPLAIN VERBOSE** cannot be displayed graphically. Click on a node icon on the **Graphical** tab to review information about that item; a popup window will display on the right side with the information about the selected object. For information on JIT statistics, triggers and a summary, Click on the button top-right corner; a similar popup window will be displayed when appropriate.

Use the download button on top left corner of the **Explain** canvas to download the plan as an SVG file.

**Note:** Download as SVG is not supported on Internet Explorer.



Note that the query plan that accompanies the **Explain analyze** is available on the **Data Output** tab.

- Table

**Table** tab shows the plan details in table format, it generates table format similar to [explain.depsez.com](http://explain.depsez.com). Each row of the table represent the data for a **Explain Plan Node**. It may contain the node information, exclusive timing, inclusive timing, actual vs planned rows differences, actual rows, planned rows, loops.

background color of the exclusive, inclusive, and Rows X columns may vary based on the difference between actual vs planned.

If percentage of the exclusive/inclusive timings of the total query time is: > 90 - Red color > 50 - Orange (between red and yellow) color > 10 - Yellow color

If planner mis-estimated number of rows (actual vs planned) by 10 times - Yellow color 100 times - Orange (between Red and Yellow) color 1000 times - Red color



Query Editor Query History Scratch Pad

```

1 SELECT DISTINCT dep.deptype, dep.refclassid, cl.relkind, ad.adbin,
2 CASE WHEN cl.relkind IS NOT NULL THEN cl.relkind || COALESCE(dep.refobjsubid::character varying, '')
3 WHEN tg.oid IS NOT NULL THEN 't':text
4 WHEN ty.oid IS NOT NULL AND ty.typbasetype = 0 THEN 'y':text
5 WHEN ty.oid IS NOT NULL AND ty.typbasetype != 0 THEN 'd':text
6 WHEN ns.oid IS NOT NULL THEN 'n':text
7 WHEN pr.oid IS NOT NULL AND prtyp.typname = 'trigger' THEN 't':text
8 WHEN pr.oid IS NOT NULL THEN 'P':text
9 WHEN la.oid IS NOT NULL THEN 'l':text
10 WHEN rw.oid IS NOT NULL THEN 'R':text
11 WHEN co.oid IS NOT NULL THEN 'C':text || contype

```

Data Output Explain Messages Notifications

Graphical Analysis Statistics

| #  | Node  | Timings   |           | Rows   |       |
|----|---|-----------|-----------|--------|-------|
|    |   | Exclusive | Inclusive | Actual | Loops |
| 1. | → Unique (actual=56.576..58.105 rows=3510 loops=1)  | 1.203 ms  | 58.105 ms | 3510   | 1     |
| 2. | → Sort (actual=56.574..56.902 rows=5340 loops=1)  | 9.964 ms  | 56.902 ms | 5340   | 1     |
| 3. | → Nested Loop Left Join (actual=13.766..46.939 rows=5340 loops=1)   | 5.186 ms  | 46.939 ms | 5340   | 1     |
| 4. | → Hash Inner Join (actual=13.751..41.754 rows=5340 loops=1)<br>Hash Cond: (dep.refclassid = pg_class.oid)                         | 1.714 ms  | 41.754 ms | 5340   | 1     |
| 5. | → Hash Left Join (actual=13.599..39.904 rows=7787 loops=1)<br>Hash Cond: (pr.proretype = prtyp.oid)                               | 2.009 ms  | 39.904 ms | 7787   | 1     |
| 6. | → Hash Left Join (actual=13.476..37.783 rows=7787 loops=1)<br>Hash Cond: (dep.refobjid = fdw.oid)                                 | 1.703 ms  | 37.783 ms | 7787   | 1     |
| 7. | → Hash Left Join (actual=13.451..36.079 rows=7787 loops=1)<br>Hash Cond: (dep.refobjid = fs.oid)                                  | 1.733 ms  | 36.079 ms | 7787   | 1     |
| 8. | → Hash Left Join (actual=13.44..34.345 rows=7787 loops=1)<br>Hash Cond: ((att.attrelid = ad.adrelid) AND (att.attnum = ad.adnum)) | 1.84 ms   | 34.345 ms | 7787   | 1     |
| 9. | → Merge Left Join (actual=13.42..32.497 rows=7787 loops=1)  | 1.537 ms  | 32.497 ms | 7787   | 1     |

- Statistics

Statistics tab shows two tables: 1. Statistics per Plan Node Type 2. Statistics per Table

Query Editor Query History Scratch Pad

```

1 SELECT DISTINCT dep.deptype, dep.refclassid, cl.relkind, ad.adbin,
2 CASE WHEN cl.relkind IS NOT NULL THEN cl.relkind || COALESCE(dep.refobjsubid::character varying, '')
3 WHEN tg.oid IS NOT NULL THEN 't':text
4 WHEN ty.oid IS NOT NULL AND ty.typbasetype = 0 THEN 'y':text
5 WHEN ty.oid IS NOT NULL AND ty.typbasetype != 0 THEN 'd':text
6 WHEN ns.oid IS NOT NULL THEN 'n':text
7 WHEN pr.oid IS NOT NULL AND prtyp.typname = 'trigger' THEN 't':text
8 WHEN pr.oid IS NOT NULL THEN 'P':text
9 WHEN la.oid IS NOT NULL THEN 'l':text
10 WHEN rw.oid IS NOT NULL THEN 'R':text
11 WHEN co.oid IS NOT NULL THEN 'C':text || contype

```

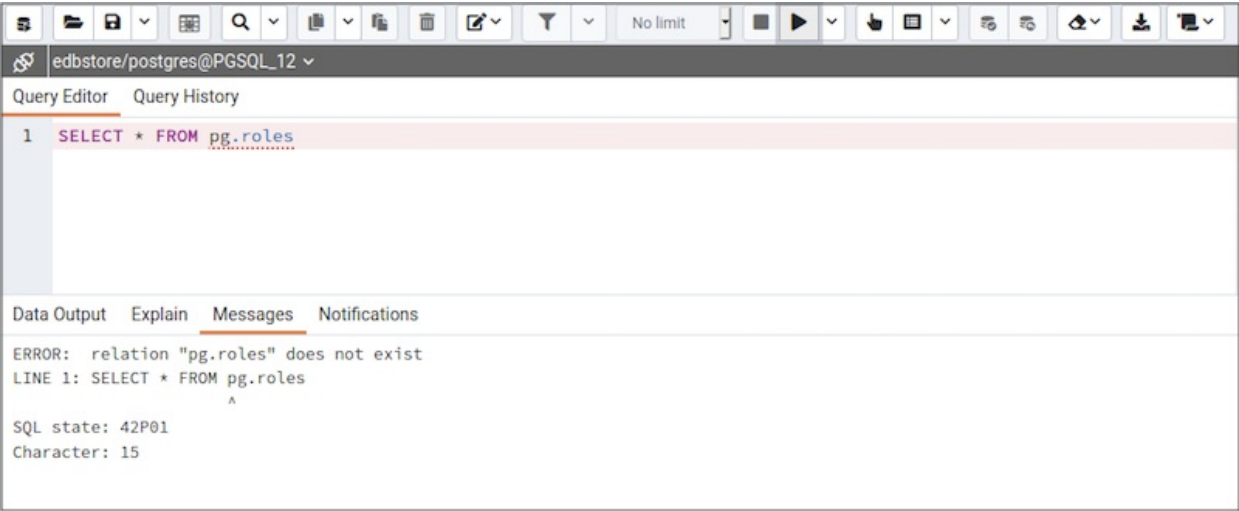
Data Output Explain Messages Notifications

Graphical Analysis Statistics

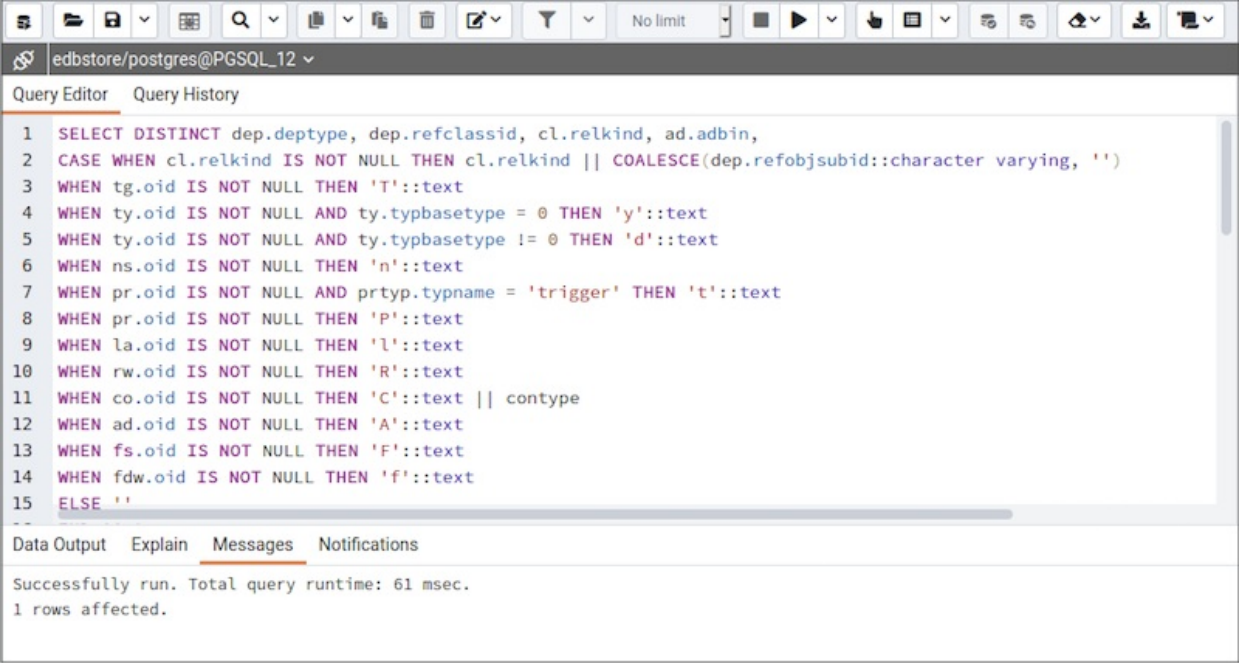
| Statistics per Node Type |       |            |            | Statistics per Table |            |              |            |
|--------------------------|-------|------------|------------|----------------------|------------|--------------|------------|
| Node type                | Count | Time spent | % of query | Table name           | Scan count | Total time   | % of query |
| Hash                     | 12    | 3.164 ms   | 5.45%      | Node type            | Count      | Sum of times | % of table |
| Hash Inner Join          | 1     | 1.714 ms   | 2.95%      | pg_attrdef           | 1          | 0.004 ms     | 0.01%      |
| Hash Left Join           | 10    | 12.938 ms  | 22.27%     | Seq Scan             | 1          | 0.004 ms     | 100%       |
| Hash Right Join          | 1     | 0.062 ms   | 0.11%      | pg_attribute         | 1          | 0.593 ms     | 1.03%      |
| Index Only Scan          | 1     | 0 ms       | 0%         | Seq Scan             | 1          | 0.593 ms     | 100%       |
| Index Scan               | 3     | 0.065 ms   | 0.12%      | pg_class             | 4          | 0.235 ms     | 0.41%      |
| Materialize              | 4     | 0.027 ms   | 0.05%      | Index Scan           | 1          | 0.038 ms     | 16.18%     |
| Merge Left Join          | 7     | 11.159 ms  | 19.21%     | Seq Scan             | 3          | 0.197 ms     | 83.83%     |
| Nested Loop Left Join    | 4     | 12.214 ms  | 21.03%     | pg_constraint        | 1          | 0.022 ms     | 0.04%      |
| Seq Scan                 | 20    | 2.215 ms   | 3.82%      | Index Scan           | 1          | 0.022 ms     | 100%       |
| Sort                     | 7     | 13.36 ms   | 23%        | pg_depend            | 1          | 0.586 ms     | 1.01%      |
| Unique                   | 1     | 1.203 ms   | 2.08%      | Seq Scan             | 1          | 0.586 ms     | 100%       |

## Messages Panel

Use the **Messages** tab to view information about the most recently executed query:



If the server returns an error, the error message will be displayed on the **Messages** tab, and the syntax that caused the error will be underlined in the SQL editor. If a query succeeds, the **Messages** tab displays how long the query took to complete and how many rows were retrieved:



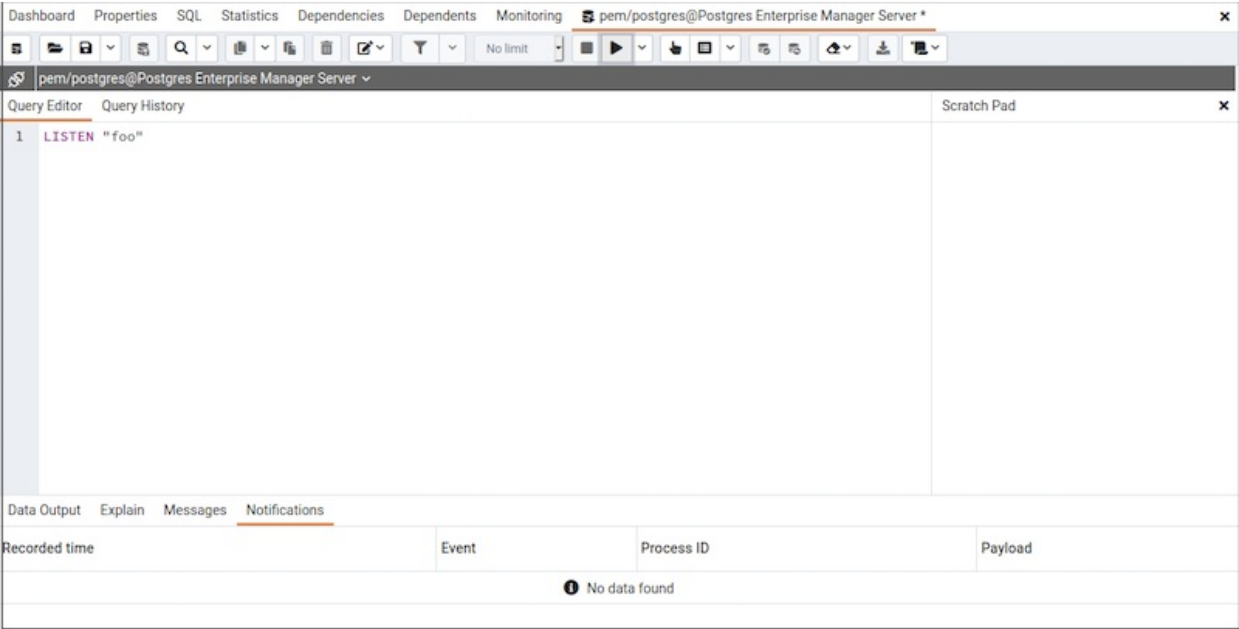
*Query tool output information*

**Notifications Panel**

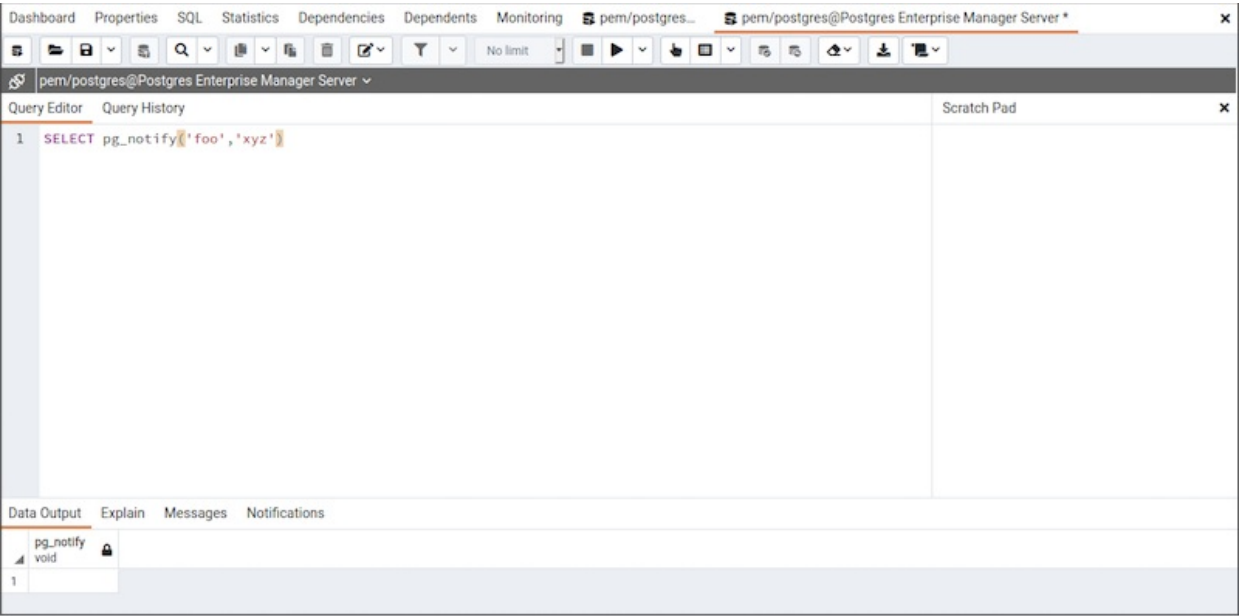
Use the **Notifications** tab to view the notifications using PostgreSQL **Listen/ Notify** feature. For more details see [PostgreSQL documentation] ([https:// www.postgresql.org/docs/current/sql-listen.html](https://www.postgresql.org/docs/current/sql-listen.html)).

Example:

- 1. Execute `LISTEN "foo"` in first **Query tool** session



2. In the another `Query tool` session, execute `Notify` command or `pg_notify` function to send the notification of the event together with the payload.



3. You can observe the `Notification` tab in the first `Query tool` session where it shows the Recorded time, Event, Process ID, and the Payload of the particular channel.

DashboardPropertiesSQLStatisticsDependenciesDependentsMonitoringpem/postgres@Postgres Enterprise Manager Server \*pem/postgres...

pem/postgres@Postgres Enterprise Manager Server

Query EditorQuery HistoryScratch Pad

1LISTEN "foo"

Data OutputExplainMessagesNotifications

| Recorded time              | Event | Process ID | Payload |
|----------------------------|-------|------------|---------|
| 2021-05-10 17:02:49.007598 | foo   | 91101      | xyz     |
| 2021-05-10 17:03:01.070535 | foo   | 91101      | xyz     |
| 2021-05-10 17:03:11.074886 | foo   | 91101      | xyz     |

Query History Panel

Use the **Query History** tab to review activity for the current session:

edbstore/postgres@PGSQL\_12

Query EditorQuery History

Show queries generated internally by Postgres Enterprise Manager?  
☒ Yes

Today - 11/17/2020

SELECT DISTINCT dep.deptype, dep.refclassid, cl.r...

15:42:01

SELECT DISTINCT dep.deptype, dep.refclassid, cl.rel...

15:34:04

SELECT \* FROM pg\_roles

15:23:25

SELECT \* FROM pg\_roles

15:23:11

SELECT \* FROM pg\_database

15:21:09

select empno, ename, deptno as department from edbu...

15:11:31

11/17/2020 3:42:01 PM161 msec

DateRows AffectedDuration

CopyCopy to Query Editor

SELECT DISTINCT dep.deptype, dep.refclassid, cl.relkind  
CASE WHEN cl.relkind IS NOT NULL THEN cl.relkind || CO  
WHEN tg.oid IS NOT NULL THEN 'T':text  
WHEN ty.oid IS NOT NULL AND ty.typbasetype = 0 THEN 'y  
WHEN ty.oid IS NOT NULL AND ty.typbasetype != 0 THEN '  
WHEN ns.oid IS NOT NULL THEN 'n':text  
WHEN pr.oid IS NOT NULL AND prtyp.typname = 'trigger'

Messages  
Successfully run. Total query runtime: 61 msec.  
1 rows affected.

The Query History tab displays information about recent commands:

- The date and time that a query was invoked.
- The text of the query.
- The number of rows returned by the query.
- The amount of time it took the server to process the query and return a result set.
- Messages returned by the server (not noted on the **Messages** tab).
- The source of the query (indicated by icons corresponding to the toolbar).

You can show or hide the queries generated internally by PEM (during 'View/Edit Data' or 'Save Data' operations).

To erase the content of the **Query History** tab, select **Clear history** from the **Clear** drop-down menu.

Query History is maintained across sessions for each database on a per-user basis when running in Query tool mode. In View/Edit Data mode, history is not

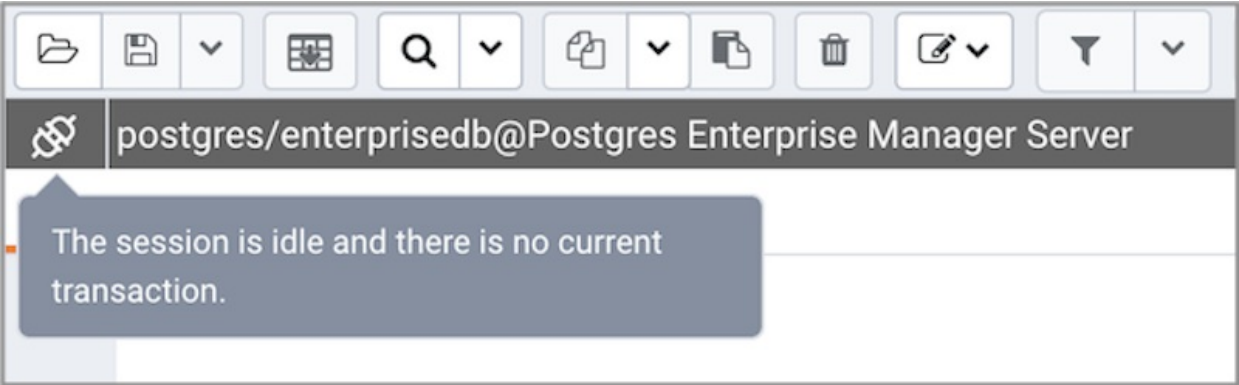
Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

870

retained. By default, the last 20 queries are stored for each database. This can be adjusted in `config_local.py` by overriding the `MAX_QUERY_HIST_STORED` value.

Connection Status

Use the `Connection status` feature to view the current connection and transaction status by clicking on the status icon in query tool:



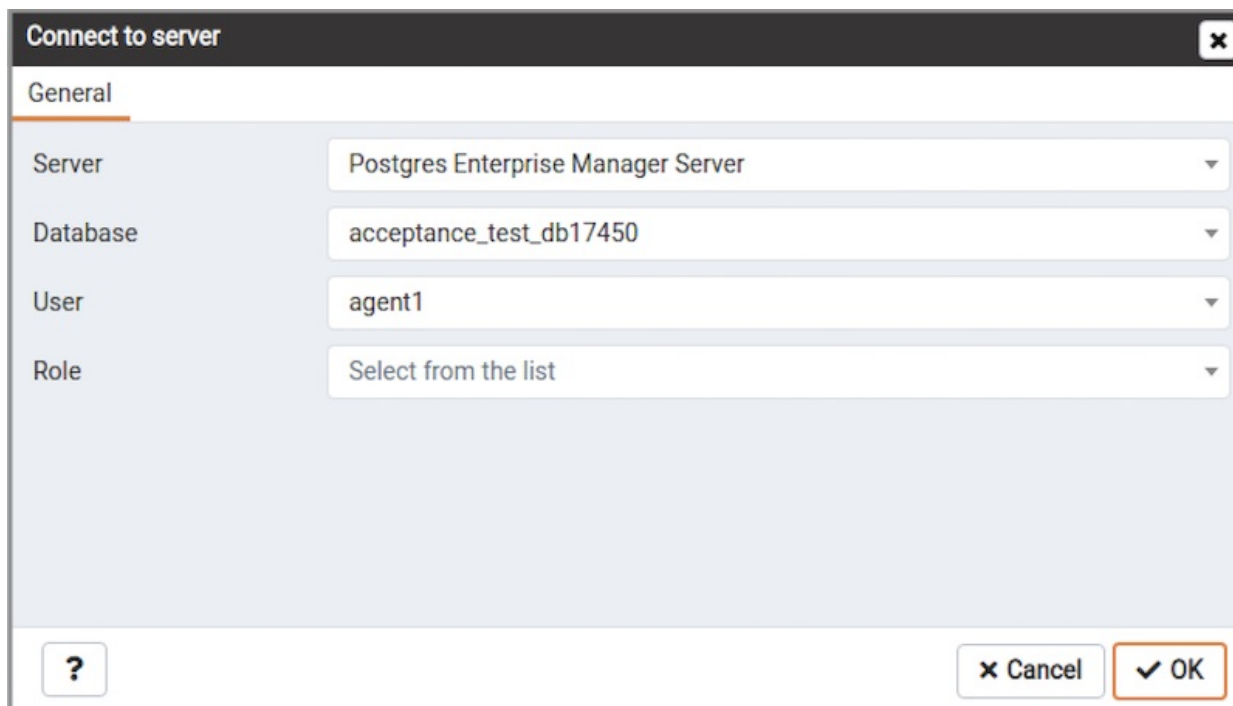
Change connection

User can connect to another server or database from existing open session of query tool.

- Click on the connection link next to connection status.
- Now click on the `<New Connection>` option from the dropdown.



- Now select server, database, user, and role to connect and click OK.



**Connect to server**

General

Server: Postgres Enterprise Manager Server

Database: acceptance\_test\_db17450

User: agent1

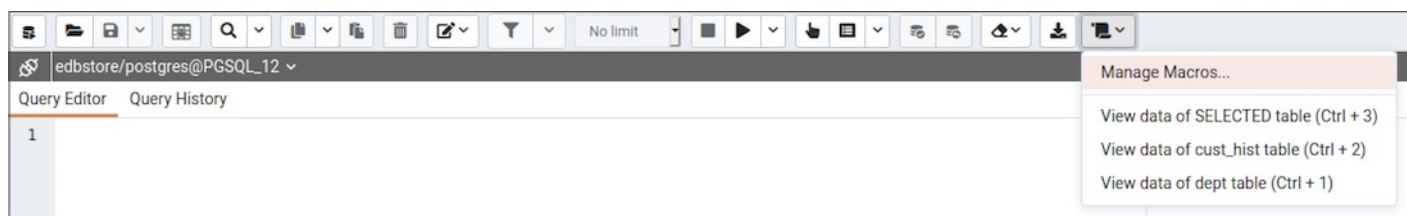
Role: Select from the list

? Cancel OK

- A newly created connection will now get listed in the options.
- To connect, select the newly created connection from the dropdown list.

## Macros

Query tool Macros enable you to execute pre-defined SQL queries with a single key press. Pre-defined queries can contain the placeholder \$SELECTION\$. Upon macro execution, the placeholder will be replaced with the currently selected text in the Query Editor pane of the Query tool.



To create a macro, select the *Manage Macros* option from the *Macros* menu on the *Query tool*. Select the key you wish to use, enter the name of the macro, and the query, optionally including the selection placeholder, and then click the *Save* button to store the macro.



Manage Macros

| Key      | Name                         | SQL  |
|----------|------------------------------|--|
| Ctrl + 1 | View data of dept table      | SELECT * FROM edbuser.dept;                            |
| Ctrl + 2 | View data of cust_hist table | SELECT * FROM edbuser.cust_hist WHERE customerid=7888; |
| Ctrl + 3 | View data of SELECTED table  | SELECT * FROM \$SELECTION\$;                           |
| Ctrl + 4 |                              |  |
| Ctrl + 5 |                              |  |
| Ctrl + 6 |                              |  |
| Ctrl + 7 |                              |  |
| Ctrl + 8 |                              |  |
| Ctrl + 9 |                              |  |
| Ctrl + 0 |                              |  |
| Alt + F1 |                              |  |

?

CancelSave

To clear a macro, select the macro on the *Manage Macros* dialogue, and then click the *Clear* button.

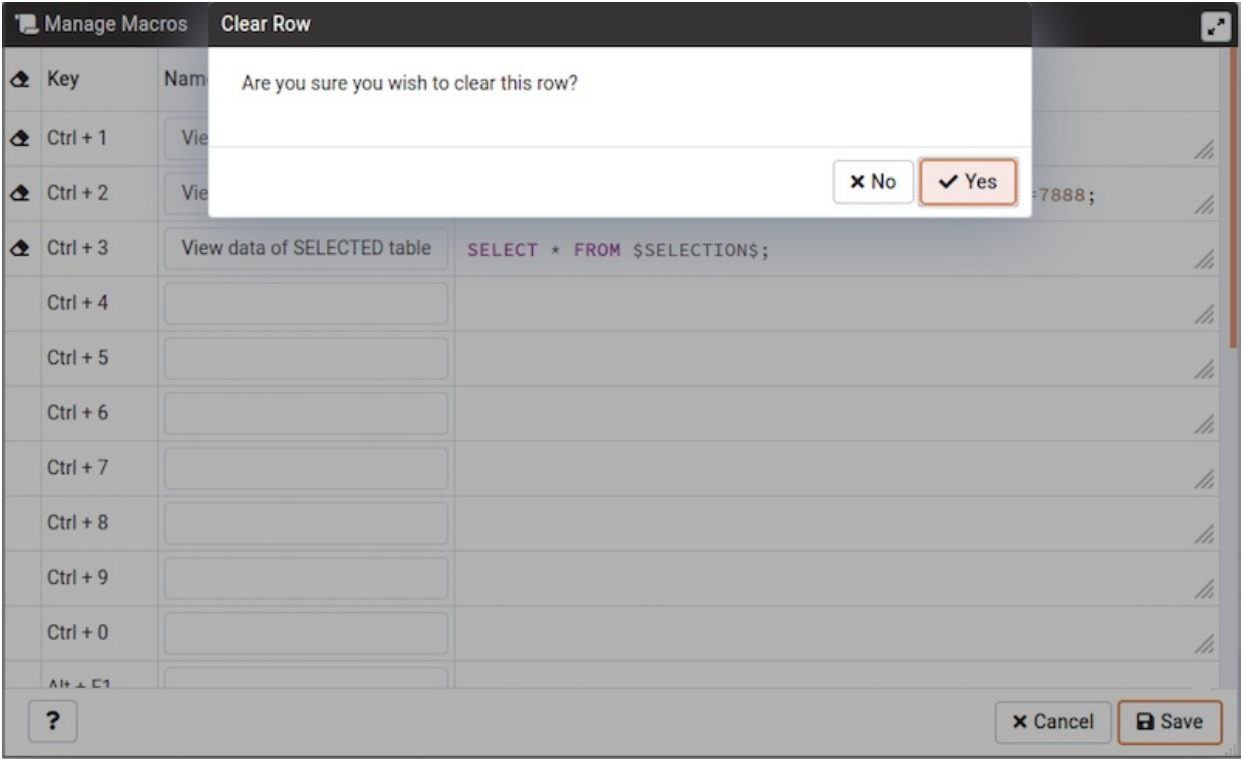
Manage Macros

|          |                              |  |
|----------|------------------------------|--|
| Ctrl + 1 | View data of dept table      | SELECT * FROM edbuser.dept;                            |
| Ctrl + 2 | View data of cust_hist table | SELECT * FROM edbuser.cust_hist WHERE customerid=7888; |
| Ctrl + 3 | View data of SELECTED table  | SELECT * FROM \$SELECTION\$;                           |
| Ctrl + 4 |                              |  |
| Ctrl + 5 |                              |  |
| Ctrl + 6 |                              |  |
| Ctrl + 7 |                              |  |
| Ctrl + 8 |                              |  |
| Ctrl + 9 |                              |  |
| Ctrl + 0 |                              |  |
| Alt + F1 |                              |  |

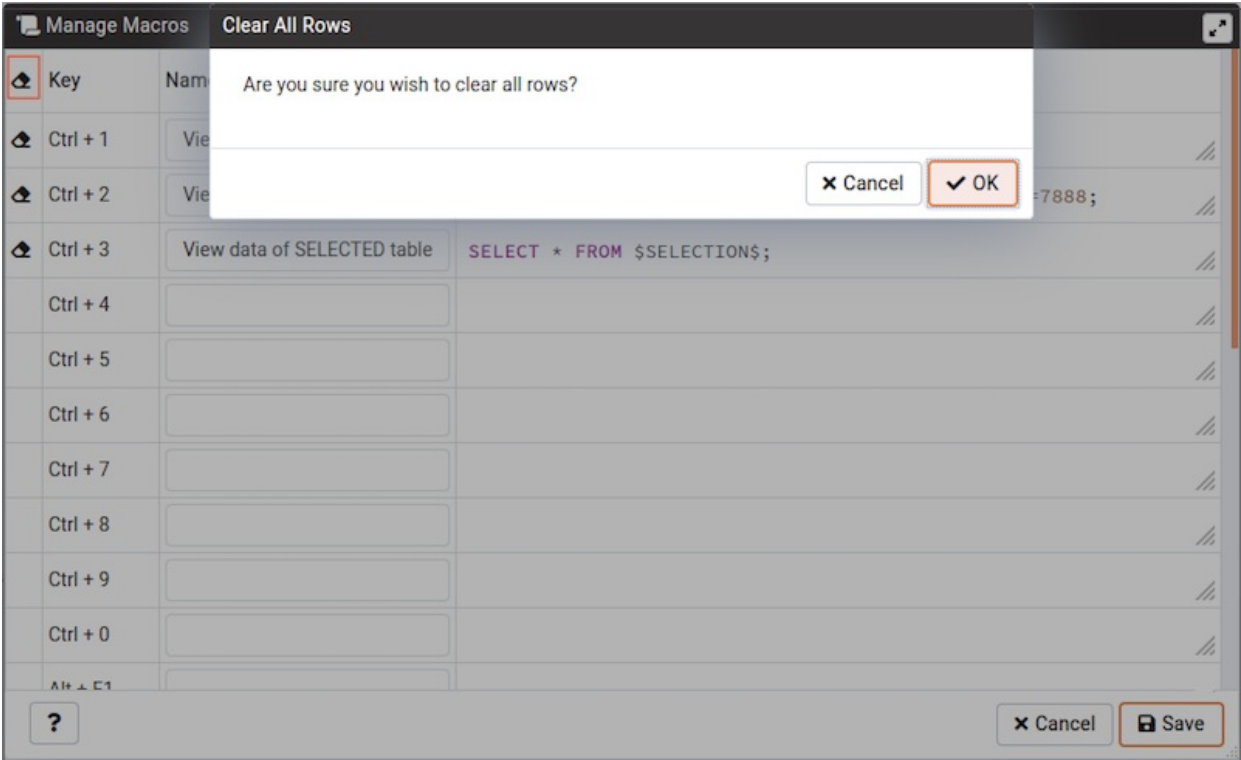
?

ClearCancelSave

The server will prompt you for confirmation to clear the macro.

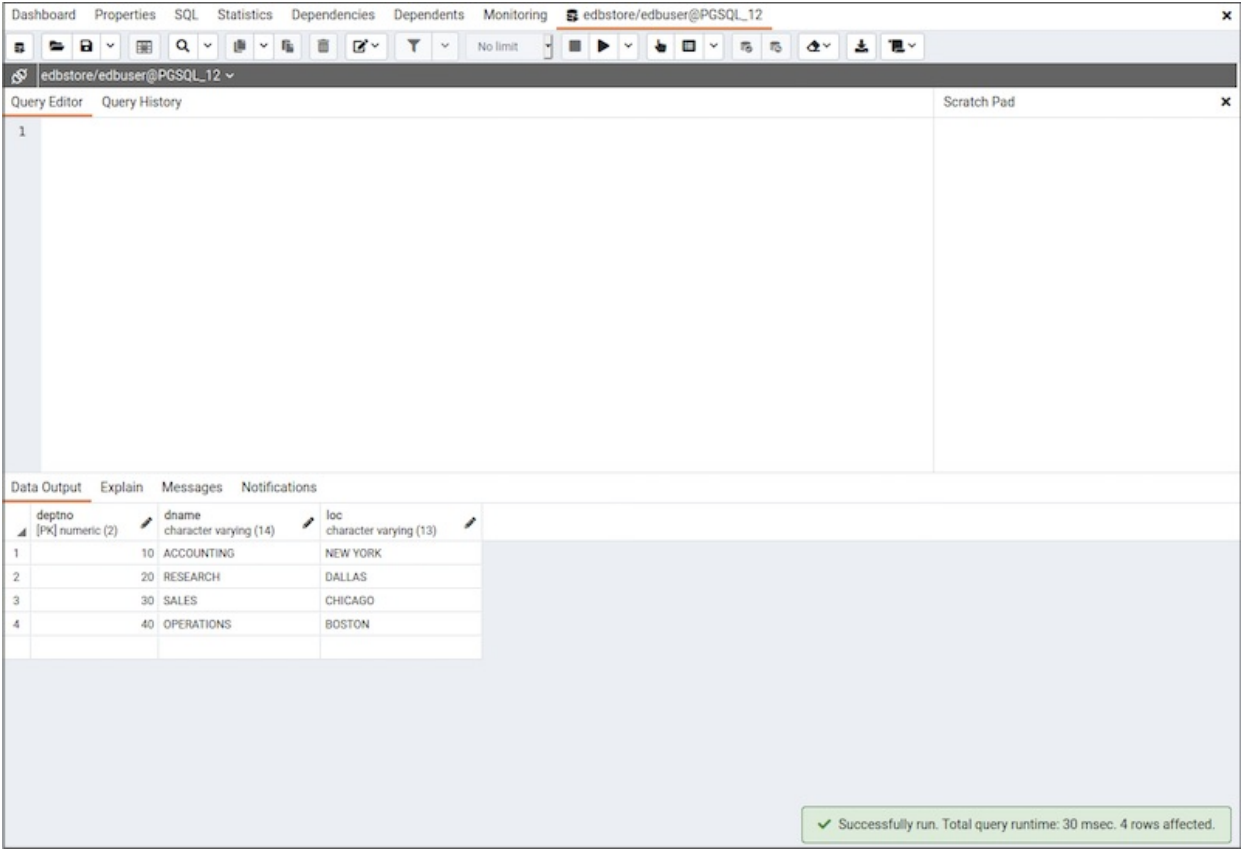


To clear all macros, click on the *Clear* button on left side of the key. The server will prompt you for confirmation to clear all the rows.



To execute a macro, simply select the appropriate shortcut keys, or select it from the *Macros* menu.





Note




If `ENABLE_DATA_ACCESS_TOOLS` configuration option is set to False then Query tool is disabled.




























### 34.8.3 Interpreting Graphical Query Plans

The graphical explain plan provides clues that can help you identify the aspects of the selected query that consume the most resources; within the diagram, thicker lines indicate the portions of the query that are expected to take the most processing time.

To view a graphical interpretation of an executed query, select `Explain` or `Explain Analyze` from the `Execute/Refresh` drop-down menu. Please note that you can use the `Explain Options` pull-aside menu to specify the level of detail displayed for each node.

The screenshot displays the DBeaver SQL Editor interface. At the top, there are tabs for 'Data Output', 'Explain' (which is selected), 'Messages', and 'Notifications'. Below these are sub-tabs for 'Graphical', 'Analysis', and 'Statistics'. The main area shows a query plan for a SQL query. The query is:   
 SELECT \* FROM server\_info, agent\_server\_binding, probe   
 WHERE server\_info.parent\_node = agent\_server\_binding.parent\_node   
 AND agent\_server\_binding.parent\_node = probe.parent\_node   
 AND agent\_server\_binding.\_serial = 15   
 The query plan consists of several nodes:   
 1. 'server\_info' (Table Scan)   
 2. 'agent\_server\_binding' (Seq Scan)   
 3. 'probe' (Table Scan)   
 4. 'Hash' (Hash)   
 5. 'Materialize' (Materialize)   
 6. 'Nested Loop Inner Join'   
 7. 'Hash Inner Join'   
 8. 'Hash Inner Join'   
 A tooltip is visible over the 'agent\_server\_binding' node, showing its properties:   
 parse: true   
 Node Type: Seq Scan   
 Parent Relationship: Outer   
 Parallel Aware: false   
 Relation Name: agent\_server\_binding   
 Alias: b   
 total\_time: 1\_2\_1\_1\_1\_1   
 parent\_node: 15   
 \_serial: 15   
 The interface also includes icons for search, zoom, and download at the top left.

| Icon   | Represents       | Description  |
|--|------------------|--|
|  | Result Set       | The Result Set icon represents a simple result set in the query plan. Typically, a Result Set operator is used to evaluate a query that does not fetch data from a table.  |
|  | Aggregate        | The server creates an Aggregate operator whenever the query invokes an aggregate function (a function that returns a value computed from multiple rows). Aggregate functions include: AVG(), COUNT(), MAX(), MIN(), STDDEV(), SUM(), and VARIANCE(). |
|  | Window Aggregate | The server may use a Window aggregate operator to implement windowed aggregate functions; a windowed aggregate function is a function that returns a value computed from a set of rows within the input.   |

| Icon   | Represents                              | Description   |
|--|---|---|
|    | Seek                                    | The server may use a Seek operator in any plan that includes an Index Scan operator. The Seek operator represents a probe into the heap to fetch the tuple that corresponds to an index entry (found by the Index Scan operator). |
|    | Seq Scan                                | The server may use a Seq scan (sequential scan) to read through a table from beginning to end.  |
|    | Index Scan                              | The server may use an Index scan operator to read through a table in the order specified in the index.  |
|    | CTE Scan                                | The server may use a CTE Scan operator if the query performs a scan of a common table expression.   |
|    | Tuple ID Scan                           | The server uses a Tuple ID scan if the query uses the Tuple ID (ctid) as a constraint in a WHERE clause.  |
|    | Group                                   | The server may use a Group operator when the query includes a GROUP BY clause. The operator requires a single input set ordered by the target column(s).  |
|    | Sort                                    | The server may use a Sort operator when a query includes an ORDER BY clause to impose an order on a result set.   |
|    | Limit                                   | The server may use the Limit operator to limit the size of a result set (when a query includes the LIMIT or OFFSET clause).   |
|    | Sub Plan                                | The server may use a Subplan operator for queries that include subselects.  |
|    | Unique                                  | The server may use the Unique operator to remove duplicate values from a result set; the diagram will include a Unique operator if the query includes a DISTINCT clause.  |
|    | Hash                                    | The server may use a Hash operator when joining two input sets that are not ordered by the column that controls the join.   |
|    | Hash Semi-Join                          | The server may use a Hash Semi-Join operator to evaluate a query that joins two tables, but returns data from only one of those tables.   |
|    | Hash Anti-Join                          | The server may use a Hash Anti-Join operator to evaluate a query that includes a NOT IN clause.   |
|   | Anti-Join                               | The server may use an Anti-Join operator to evaluate a query that includes a NOT IN clause.   |
|  | Join                                    | The server may use a Join operator when joining two input sets that are ordered by the column that controls the join.   |
|  | Recursive Union                         | The server may use a Recursive Union operator if the query includes a WITH RECURSIVE clause.  |
|  | Set Operator                            | The server may use a Set operator if the query includes an INTERSECT, INTERSECT ALL, EXCEPT or EXCEPT ALL clause.   |
|  | Hash Set Operator (Setop) Intersect     | The server may use an Intersect Set operator if the query includes an INTERSECT clause.   |
|  | Hash Set Operator (Setop) Intersect All | The server may use an Intersect Set operator if the query includes an INTERSECT ALL.  |
|  | Hash Set Operator (Setop) Except        | The server may use an Except Set operator if the query includes an EXCEPT.  |
|  | Hash Set Operator (Setop) Except All    | The server may use an Except All Set operator if the query includes an EXCEPT ALL clause.   |
|  | Materialize                             | The server may choose to use a Materialize operator for a subselect operation (a nested query).   |
|  | Append                                  | The server may use an Append operator to implement queries that contain a UNION clause.   |
|  | Nested Loop                             | The server may use a Nested Loop operator to perform a join between two tables. When implementing a nested loop, the server searches for rows from the inner table that match the corresponding row in the outer table.           |
|  | Merge Join                              | The server may use a Merge Join operator to join two tables. A Merge Join requires two sets of inputs, where each set is ordered by the column used for the comparison.   |
|  | Merge Semi-Join                         | The server may use a Merge Semi-Join operator to evaluate a query that joins two tables, but returns data from only one of those tables.  |
|  | Merge Anti-Join                         | The server may use a Merge Anti-Join operator to evaluate a query that includes a NOT IN clause.  |

- For more information about interpreting and understanding a query plan, see Using EXPLAIN, in the PostgreSQL documentation.
- For information about using PEM's Index Advisor, see the EDB Postgres Advanced Server Guide, available from the EnterpriseDB website at [www.enterprisedb.com](http://www.enterprisedb.com).
- For information about using configuration parameters to influence query plans, see Query Planning, in the PostgreSQL documentation.
- For more information about using Oracle-compatible optimizer hints, see the EDB Postgres Advanced Server Oracle Compatibility Developer's Guide, available from the EnterpriseDB website at [www.enterprisedb.com](http://www.enterprisedb.com).

To review or modify data, right click on a table or view name in the **Browser** tree control. When the context menu opens, use the **View/Edit Data** menu to specify the number of rows you would like to display in the editor panel.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

To modify the content of a table, each row in the table must be uniquely identifiable. If the table definition does not include an OID or a primary key, the displayed data is read only. Note that views cannot be edited; updatable views (using rules) are not supported.

The editor features a toolbar that allows quick access to frequently used options, and a work environment divided into two panels:

- The upper panel displays the SQL command that was used to select the content displayed in the lower panel.
- The lower panel (the Data Grid) displays the data selected from the table or view.

The View/Edit Data Toolbar

The toolbar includes context-sensitive icons that provide shortcuts to frequently performed tasks.



Hover over an icon to display a tooltip that describes the icon's functionality.

| Icon            | Behavior  | Sh<br>ort<br>cu<br>t           |
|-----------------|---|--------------------------------|
| Save            | Use the <b>Save</b> icon to save your changes to the currently displayed table contents.  |                                |
| Find            | Use options on the <b>Find</b> menu to access Search and Replace functionality or to Jump to another line.  | Ct<br>rl/<br>C<br>m<br>d<br>+F |
| Copy            | Click the <b>Copy</b> icon to copy the currently selected data.   | Ct<br>rl+<br>C                 |
| Paste Row       | Click the <b>Paste Row</b> icon to paste the content that is currently on the clipboard.  |                                |
| Delete Row      | Use the <b>Delete Row</b> icon to delete all the selected rows from the output panel.   |                                |
| Filter          | <p>Click the <b>Filter</b> icon to open a dialog that allows you to write and apply a filter for the content currently displayed in the output panel. Click the down arrow to open the <b>Filter</b> drop-down menu and select from pre-defined options:</p> <p>Use options on the <b>Filter</b> menu to quick-sort or quick-filter the data set:</p> <ul style="list-style-type: none"><li>- Filter: This option opens a dialog that allows you to define a filter. A filter is a condition that is supplied to an arbitrary WHERE clause that restricts the result set.</li><li>- Remove Filter: This option removes all selection / exclusion filter conditions.</li><li>- By Selection: This option refreshes the data set and displays only those rows whose column value matches the value in the cell currently selected.</li><li>- Exclude Selection: This option refreshes the data set and excludes those rows whose column value matches the value in the cell currently selected.</li></ul> |                                |
| No limit        | Use the <b>No limit</b> drop-down listbox to specify how many rows to display in the output panel. Select from: <b>No limit</b> (the default), <b>1000 rows</b> , <b>500 rows</b> , or <b>100 rows</b> .  |                                |
| Execute/Refresh | Click the <b>Execute/Refresh</b> icon to execute the SQL command that is displayed in the top panel. If you have not saved modifications to the content displayed in the data grid, you will be prompted to confirm the execution. To preserve your changes before refreshing the content, click the <b>Save</b> toolbar button before executing the refresh.   | F5                             |
| Stop            | Click the <b>Stop</b> icon to cancel the execution of the currently running query.  |                                |

| Icon   | Behavior   | Short cut |
|--|--|-----------|
|  | Use the <b>Clear History</b> drop-down menu to erase the contents of the <b>History</b> tab.   |           |
|  | Click the <b>Download as CSV</b> icon to download the result set of the current query to a comma-separated list. You can control the CSV settings through <b>Preferences -&gt; SQL Editor -&gt; CSV output</b> dialogue. | F8        |

The Data Grid

The top row of the data grid displays the name of each column, the data type, and if applicable, the number of characters allowed. A column that is part of the primary key will additionally be marked with [PK].

To modify the displayed data:

- To change a numeric value within the grid, double-click the value to select the field. Modify the content in the square in which it is displayed.
- To change a non-numeric value within the grid, double-click the content to access the edit bubble. After modifying the content of the edit bubble, click the **Save** button to display your changes in the data grid, or **Cancel** to exit the edit bubble without saving.

To enter a newline character, click Ctrl-Enter or Shift-Enter. Newline formatting is only displayed when the field content is accessed via an edit bubble.

To add a new row to the table, enter data into the last (unnumbered) row of the table. As soon as you store the data, the row is assigned a row number, and a fresh empty line is added to the data grid.

To write a SQL NULL to the table, simply leave the field empty. When you store the new row, the will server fill in the default value for that column. If you store a change to an existing row, the value NULL will explicitly be written.

To write an empty string to the table, enter the special string " (two single quotes) in the field. If you want to write a string containing solely two single quotes to the table, you need to escape these quotes, by typing "

To delete a row, press the **Delete** toolbar button. A popup will open, asking you to confirm the deletion.

To commit the changes to the server, select the **Save** toolbar button. Modifications to a row are written to the server automatically when you select a different row.

Geometry Data Viewer

If PostGIS is installed, you can view GIS objects in a map by selecting row(s) and clicking the 'View Geometry' button in the column. If no rows are selected, the entire data set will be rendered:



DashboardPropertiesSQLStatisticsDependenciesDependentsMonitoringpublic.coffee\_shops/postgres/postgres@PostgreSQL 11

public.coffee\_shops/postgres/postgres@PostgreSQL 11

Query EditorQuery HistoryScratch Pad

1SELECT \* FROM public.coffee\_shops

2

Data OutputExplainMessagesNotifications

|    | idinteger | namecharacter varying (50)   | addresscharacter varying (50) | citycharacter varying (50) |
|----|-----------|------------------------------|-------------------------------|----------------------------|
| 1  | 1         | 1369 Coffee House            | 1369 Cambridge St             | Cambridge                  |
| 2  | 2         | 1369 Coffee House            | 757 Massachusetts Ave         | Cambridge                  |
| 3  | 3         | Acetuna Cafe                 | 605 W Kendall St              | Cambridge                  |
| 4  | 4         | Al's Deli Cafe-Cambridge     | 1354 Massachusetts Ave        | Cambridge                  |
| 5  | 5         | Algiers Coffee House         | 40 Brattle St # 3             | Cambridge                  |
| 6  | 6         | Area Four                    | 500 Technology Sq             | Cambridge                  |
| 7  | 7         | B & B Snack Bar              | 55 Broadway                   | Cambridge                  |
| 8  | 8         | Basha Cafe                   | 26 New St                     | Cambridge                  |
| 9  | 9         | Beantowne Coffee House       | 1 Kendall Sq # B2105          | Cambridge                  |
| 10 | 10        | Boston Tea Stop              | 54 JFK St # 1                 | Cambridge                  |
| 11 | 11        | Broadway Marketplace LLC     | 468 Broadway                  | Cambridge                  |
| 12 | 12        | Cafe Avec                    | 2269 Massachusetts Ave # 1    | Cambridge                  |
| 13 | 13        | Cafe Pamplona                | 12 Bow St                     | Cambridge                  |
| 14 | 14        | Cafe Zing                    | 25 White St                   | Cambridge                  |
| 15 | 15        | Cambridge Coffee Shop        | 847 Cambridge St              | Cambridge                  |
| 16 | 16        | Central Brew Cafe & Espresso | 350 Massachusetts Ave         | Cambridge                  |
| 17 | 17        | Conroy Commons               | 13 Applan Way                 | Cambridge                  |
| 18 | 18        | Crema Cafe                   | 27 Brattle St                 | Cambridge                  |

Geometry Viewer

You can adjust the layout by dragging the title of the panel. To view the properties of the geometries directly in map, just click the specific geometry:

DashboardPropertiesSQLStatisticsDependenciesDependentsMonitoringpublic.coffee\_shops/postgres/postgres@PostgreSQL 11

public.coffee\_shops/postgres/postgres@PostgreSQL 11

Query EditorQuery HistoryScratch Pad

1SELECT \* FROM public.coffee\_shops

2

Data OutputExplainMessagesNotificationsGeometry Viewer

id25

nameDunkin' Donuts

address1 White St

cityCambridge

stateMA

zip02140

lat42.38863

Notes:

- Supported data types: The Geometry Viewer supports 2D and 3D geometries in EWKB format including Point, LineString, Polygon MultiPoint, MultiLineString, MultiPolygon and GeometryCollection.
- SRIDs: If there are geometries with different SRIDs in the same column, the viewer will render geometries with the same SRID in the map. If

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

881

SRID=4326 the OSM tile layer will be added into the map.

- **Data size:** For performance reasons, the viewer will render no more than 100000 geometries, totaling up to 20MB.
- **Internet access:** An internet connection is required for the Geometry Viewer to function correctly.

## JSON Data Editor

A built in json editor is provided for *JSON/JSONB Data*. Double clicking on json/jsonb data type cell in data grid will open JSON Editor. Editor provides different mode to view and edit json data.



**Code Mode:** Provides way to format & compact json data. Also provides ability to repair json data by fixing quotes and escape characters, removing comments and JSONP notation and turn JavaScript objects into JSON.

**Tree Mode:** Enabled to change, add, move, remove, and duplicate fields and values. Provides ability to search & highlight data.

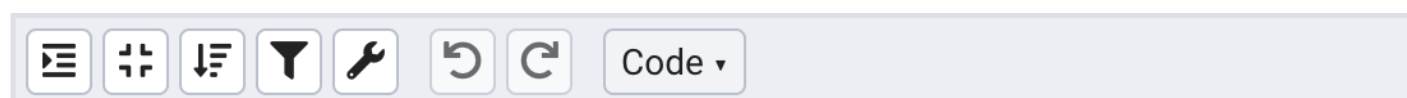
**Form Mode:** Allows only to edit values in json data there by providing ability to keep data structure unchanged while editing.

**Preview Mode\*:** Provides ability to check data before saving and also shows size of current json data. Format and compact json data as well.

## Editor Toolbar

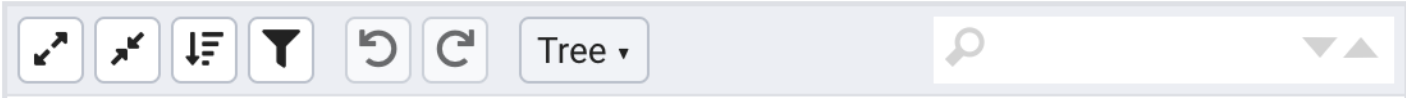
Different options are provided to manipulate json data.

Code/Preview mode:



Tree/Form mode:

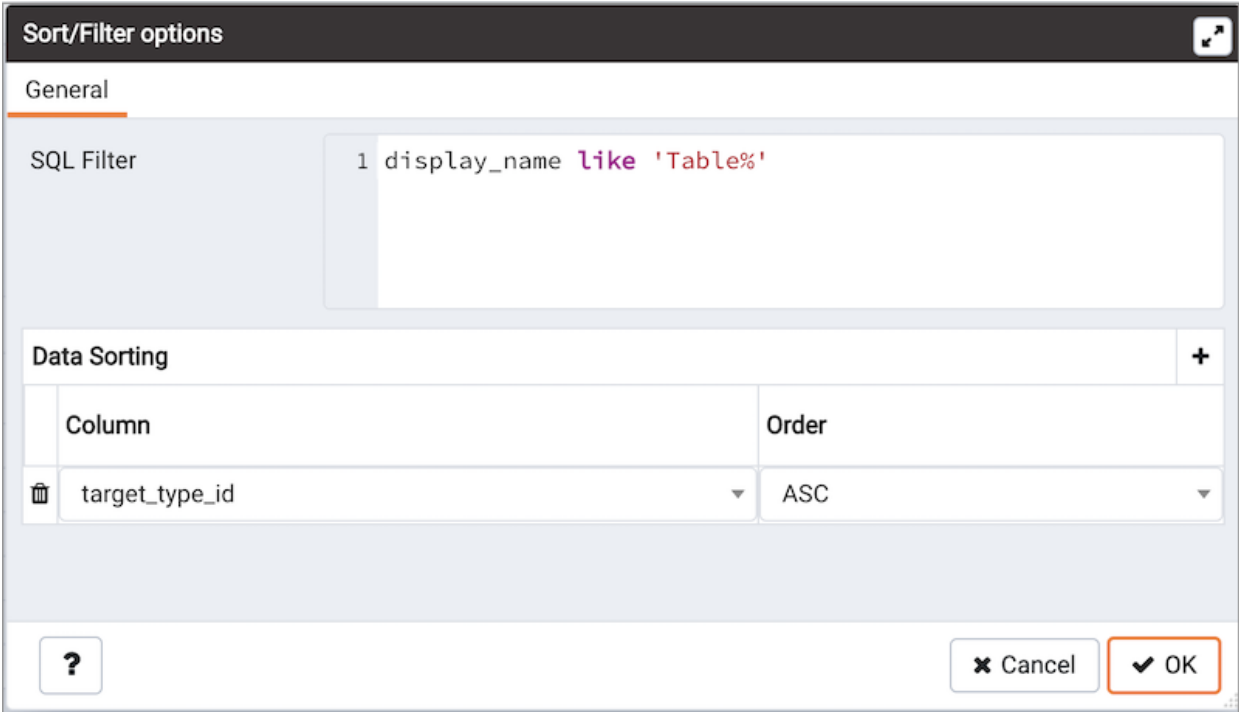




| Icon                | Behavior  | Available in mode         |
|---------------------|---|---------------------------|
| <i>Format Json</i>  | Click to <i>Format Json</i> format json data with proper indentation.   | Code, Preview             |
| <i>Compact Json</i> | Click to <i>Compact Json</i> get compacted json data.   | Code, Preview             |
| <i>Sort</i>         | Click to <i>Sort</i> icon to set sorting criteria for the data using file values.                                     | Code, Preview, Tree       |
| <i>Transform</i>    | Click to <i>Transform</i> to filter data using JSME query language.   | Code, Preview, Tree       |
| <i>Undo</i>         | Click to <i>Undo</i> to undo last action performed on data .  | Code, Preview, Tree, From |
| <i>Redo</i>         | Click to <i>Redo</i> to repat last action performed on data .   | Code, Preview, Tree, From |
| <i>Mode</i>         | Click to <i>Mode</i> dropdown to change dipaly mode of editor.Different modes avaiable are Code, Preview, Tree, From. | Code, Tree, From, Preview |
| <i>Expand All</i>   | Click to <i>Expand All</i> to expand json data.   | Tree, From                |
| <i>Collapse All</i> | Click to <i>Redo</i> to collapse json data.   | Tree, From                |
| <i>Search Box</i>   | Enter partial/complete string to search in data.  | Tree, From                |

Sort/Filter options dialog

You can access **Sort/Filter options dialog** by clicking on Sort/Filter button. This allows you to specify an SQL Filter to limit the data displayed and data sorting options in the edit grid window:



- Use **SQL Filter** to provide SQL filtering criteria. These will be added to the "WHERE" clause of the query used to retrieve the data. For example, you might enter:

```
id > 25 AND created > '2018-01-01'
```

- Use **Data Sorting** to sort the data in the output grid

To add new column(s) in data sorting grid, click on the [+] icon.

- Use the drop-down **Column** to select the column you want to sort.
- Use the drop-down **Order** to select the sort order for the column.

To delete a row from the grid, click the trash icon.

- Click the **Help** button (?) to access online help.
- Click the **Ok** button to save work.
- Click the **Close** button to discard current changes and close the dialog.

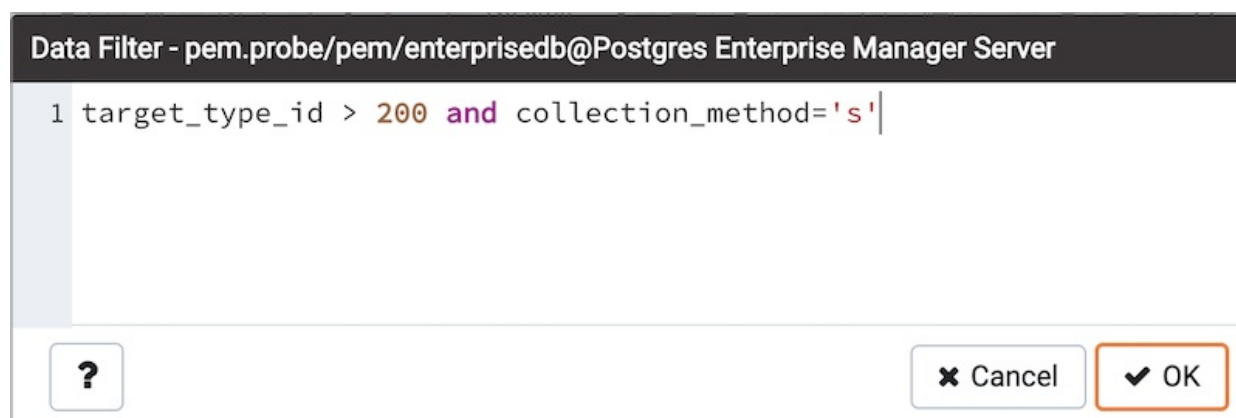
#### Note

If **ENABLE\_DATA\_ACCESS\_TOOLS** configuration option is set to False then View/Edit Data option is disabled.

### 34.8.4.1 View/Edit Data Filter

You can access **Data Filter dialog** by clicking on **Filtered Rows** toolbar button visible on the Browser panel or by selecting *View/Edit Data -> Filtered Rows* context menu option.

This allows you to specify an SQL Filter to limit the data displayed in the edit grid window:



#### Note

Use SHIFT + ENTER keys to apply filter.

### 34.8.5 Schema Diff

**Schema Diff** is a feature that allows you to compare objects between two database or two schemas. Use the **Tools** menu to access Schema Diff.

The Schema Diff feature allows you to:

- Compare and synchronize the database objects (from source to target).
- Visualize the differences between database objects.
- List the differences in SQL statement for target database objects.
- Generate synchronization scripts.

#### Note:

- The source and target database servers must be of the same major version.
- If you compare two **schemas** then dependencies won't be resolved.

Click on *Schema Diff* under the *Tools* menu to open a selection panel. To compare **databases** choose the source and target servers, and databases. To compare **schemas** choose the source and target servers, databases, and schemas. After selecting the objects, click on the *Compare* button.

You can open multiple copies of *Schema Diff* in individual tabs simultaneously. To close a copy of Schema Diff, click the \*X\* in the upper-right hand corner of the tab bar. You can rename the panel title by right-clicking and select the "Rename Panel" option.

DashboardPropertiesSQLStatisticsDependenciesDependentsMonitoringSchema Diff (Beta)

Select SourcePostgreSQL 11postgressource\_sc

Select TargetPostgreSQL 11postgrestarget\_sc

CompareGenerate ScriptFilter

Schema Objects

Comparison Result

Collations - Identical: 0 Different: 1000 Source Only: 1 Target Only: 1

|          |             |
|----------|-------------|
| abstract | Source Only |
| asset    | Target Only |
| col1     | Different   |
| col10    | Different   |
| col100   | Different   |
| col1000  | Different   |
| col101   | Different   |
| col102   | Different   |
| col103   | Different   |
| col104   | Different   |
| col105   | Different   |
| col106   | Different   |

DDL Comparison

Source

Target

Difference

Use the [Preferences](#) dialog to specify following:

- *Schema Diff* should open in a new browser tab. Set *Open in new browser tab* option to true.
- *Schema Diff* should ignore the whitespaces while comparing string objects. Set *Ignore whitespaces* option to true.
- *Schema Diff* should ignore the owner while comparing objects. Set *Ignore owner* option to true.

The *Schema Diff* panel is divided into two panels; an Object Comparison panel and a DDL Comparison panel.

The Schema Diff Object Comparison Panel

In the object comparison panel, you can select the source and target servers of the same major version, and databases to be compared. You can select any server listed under the browser tree whether it is connected or disconnected. If you select a server that is not connected then it will prompt you for the password before using the server.

Next, select the databases that will be compared. The databases can be the same or different (and within the same server or from different servers).

Dashboard Properties SQL Statistics Dependencies Dependents Monitoring Schema Diff (Beta)

Select Source PostgreSQL 11 x postgres x source\_sc x

Select Target PostgreSQL 11 x postgres x target\_sc x Compare Generate Script Filter

Select the server, database and schema for the source and target and click **Compare** to compare them.

After you select servers and databases, click on the *Compare* button to obtain the **Comparison Result**.

Dashboard Properties SQL Statistics Dependencies Dependents Monitoring Schema Diff (Beta)

Select Source PostgreSQL 11 x postgres x source\_sc x

Select Target PostgreSQL 11 x postgres x target\_sc x Compare Generate Script Filter

| Schema Objects  | Comparison Result |
|---|-------------------|
| Collations - Identical: 0 Different: 1000 Source Only: 1 Target Only: 1         |                   |
| Domains - Identical: 0 Different: 1000 Source Only: 0 Target Only: 0            |                   |
| Foreign Tables - Identical: 1000 Different: 0 Source Only: 0 Target Only: 0     |                   |
| FTS Configurations - Identical: 1000 Different: 0 Source Only: 0 Target Only: 0 |                   |
| FTS Dictionaries - Identical: 1000 Different: 0 Source Only: 0 Target Only: 0   |                   |
| fts1  | Identical         |
| fts10   | Identical         |
| fts100  | Identical         |
| fts1000   | Identical         |
| fts101  | Identical         |
| fts102  | Identical         |
| fts103  | Identical         |

Use the drop-down lists of Database Objects to view the DDL statements.

In the upper-right hand corner of the object comparison panel is a *Filter* option that you can use to filter the database objects based on the following comparison criteria:

- Identical – If the object is found in both databases with the same SQL statement, then the comparison result is identical.
- Different – If the object is found in both databases but have different SQL statements, then the comparison result is different.
- Source Only – If the object is found in source database only and not in target database, then the comparison result is source only.
- Target Only – If the object is found in target database only and not in source database, then the comparison result is target only.

Dashboard Properties SQL Statistics Dependencies Dependents Monitoring Schema Diff (Beta)

Select Source PostgreSQL 11 x postgres x source\_sc x

Select Target PostgreSQL 11 x postgres x target\_sc x Compare Generate Script Filter

| Schema Objects  | Comparison Result |
|---|-------------------|
| Collations - Identical: 0 Different: 1000 Source Only: 1 Target Only: 1 |                   |
| abstract  | Source Only       |
| asset   | Target Only       |
| col1  | Different         |
| col10   | Different         |
| col100  | Different         |
| col1000   | Different         |
| col101  | Different         |
| col102  | Different         |
| col103  | Different         |
| col104  | Different         |
| col105  | Different         |

Click on any of the database objects in the object comparison panel to display the DDL Statements of that object in the DDL Comparison panel.

Schema Diff DDL Comparison Panel

The **DDL Comparison** panel displays three columns:

- The first column displays the DDL statement of the object from the source database.
- The second column displays the DDL statement of the object from the target database.
- The third column displays the difference in the SQL statement of the target database object.

DashboardPropertiesSQLStatisticsDependenciesDependentsMonitoringSchema Diff (Beta)

Select Source

PostgreSQL 11postgressource\_sc

Select Target

PostgreSQL 11postgrestarget\_sc

Compare

Generate Script

Filter

Schema Objects

Comparison Result

Collations - Identical: 0 Different: 1000 Source Only: 1 Target Only: 1

|          |             |
|----------|-------------|
| abstract | Source Only |
| asset    | Target Only |
| col1     | Different   |
| col10    | Different   |
| col100   | Different   |
| col1000  | Different   |
| col101   | Different   |
| col102   | Different   |
| col103   | Different   |
| col104   | Different   |
| col105   | Different   |
| col106   | Different   |

DDL Comparison

Source

Target

Difference

You can review the DDL statements of all the database objects to check for the differences in the SQL statements.

Also, you can generate the SQL script of the differences found in the target database object based on the SQL statement of the source database object. To generate the script, select the checkboxes of the database objects in the object comparison panel and then click on the *Generate Script* button in the upper-right hand corner of the object comparison panel.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

887

Dashboard Properties SQL Statistics Dependencies Dependents Monitoring Schema Diff (Beta)

Select Source PostgreSQL 11 postgres source\_sc

Select Target PostgreSQL 11 postgres target\_sc Compare Generate Script Filter

Schema Objects

Comparison Result

Collations - Identical: 0 Different: 1000 Source Only: 1 Target Only: 1

|          |             |
|----------|-------------|
| abstract | Source Only |
| asset    | Target Only |
| col1     | Different   |
| col10    | Different   |
| col100   | Different   |
| col1000  | Different   |
| col101   | Different   |
| col102   | Different   |
| col103   | Different   |
| col104   | Different   |
| col105   | Different   |
| col106   | Different   |

DDL Comparison

Source

Target

Difference

1 -- Collation: coll;

2

3 -- DROP COLLATION source\_sc.coll;

4

5 CREATE COLLATION source\_sc.coll

6 (LC\_COLLATE = 'POSIX', LC\_CTYPE = 'POSIX');

7

8 ALTER COLLATION source\_sc.coll

9 OWNER TO postgres;

1 -- Collation: coll;

2

3 -- DROP COLLATION target\_sc.coll;

4

5 CREATE COLLATION target\_sc.coll

6 (LC\_COLLATE = 'C', LC\_CTYPE = 'C');

7

8 ALTER COLLATION target\_sc.coll

9 OWNER TO postgres;

1 -- WARNING:

2 -- We have found the difference in either of LC

3 -- so we need to drop the existing collation fi

4 DROP COLLATION target\_sc.coll;

5

6 CREATE COLLATION target\_sc.coll

7 (LC\_COLLATE = 'POSIX', LC\_CTYPE = 'POSIX');

Select the database objects and click on the *Generate Script* button to open the **Query tool** in a new tab, with the difference in the SQL statement displayed in the **Query Editor**.

**Note**

If `ENABLE_DATA_ACCESS_TOOLS` configuration option is set to False then **Generate Script** option is disabled.

If you have clicked on the database object to check the difference generated in the **DDL Comparison** Panel, and you have not selected the checkbox of the database object, PEM will open the **Query tool** in a new tab, with the differences in the SQL statements displayed in the **Query Editor**.

You can also use the **Copy** button to copy the difference generated in the **DDL Comparison** panel.

Dashboard Properties SQL Statistics Dependencies Dependents Monitoring Schema Diff (B... postgres/postgres@PostgreSQL 11 \*

postgres/postgres@PostgreSQL 11

Query Editor Query History Scratch Pad

1 -- This script was generated by a beta version of the Schema Diff utility in Postgres Enterprise Manager.  
2 -- This version does not include dependency resolution, and may require manual changes  
3 -- to the script to ensure changes are applied in the correct order.  
4 -- Please report an issue for any failure with the reproduction steps.  
5 BEGIN;  
6 -- WARNING:  
7 -- We have found the difference in either of LC\_COLLATE or LC\_CTYPE or LOCALE,  
8 -- so we need to drop the existing collation first and re-create it.  
9 DROP COLLATION target\_sc.coll100;  
10  
11 CREATE COLLATION target\_sc.coll100  
12 (LC\_COLLATE = 'POSIX', LC\_CTYPE = 'POSIX');  
13 END;

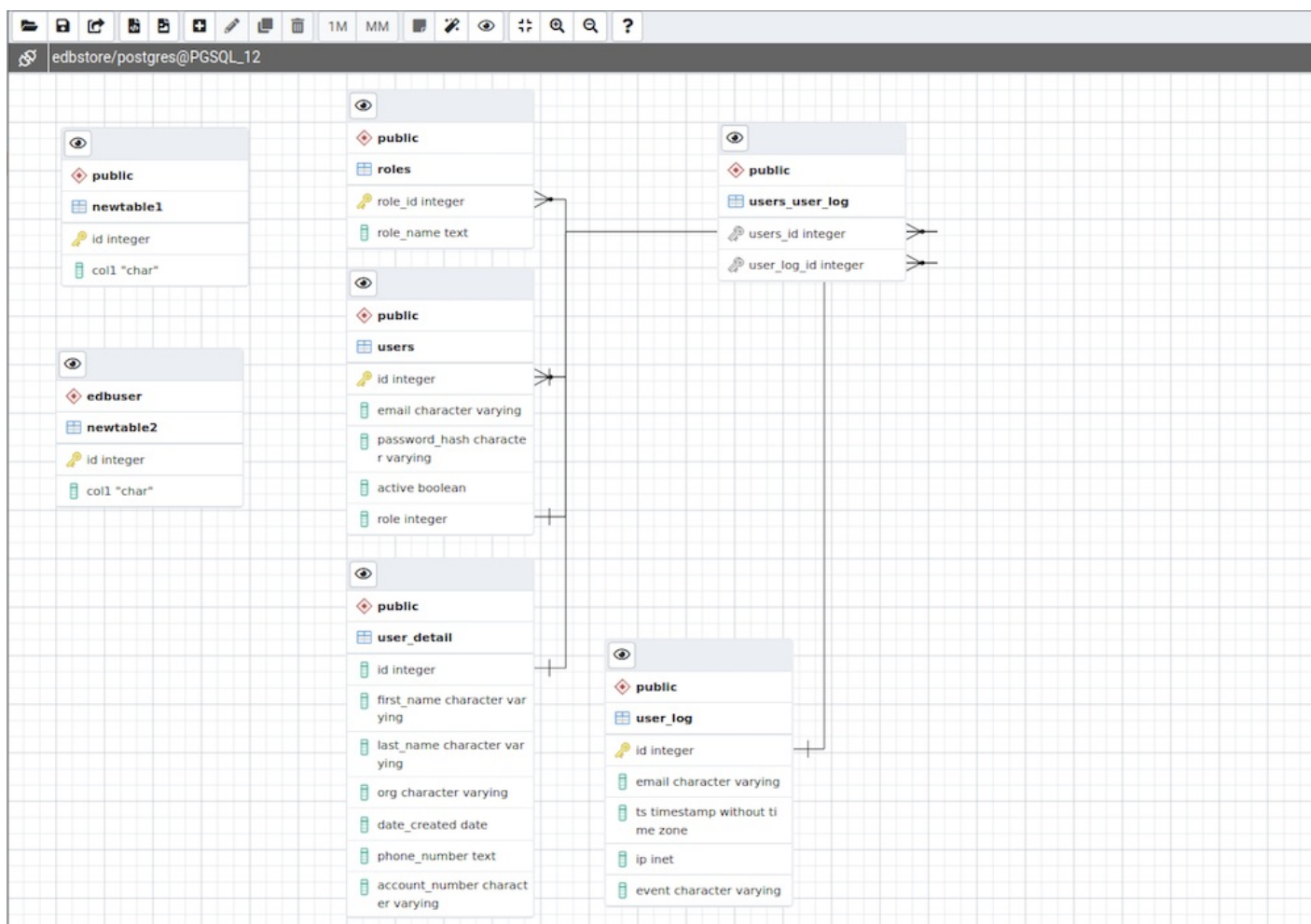
Apply the SQL Statement in the target database to synchronize the databases.

34.8.6 ERD Tool



The Entity-Relationship Diagram (ERD) tool is a database design tool that provides a graphical representation of database tables, columns, and inter-relationships. ERD can give sufficient information for the database administrator to follow when developing and maintaining the database. You can access the ERD Tool via the **ERD Tool** menu option on the **Tools** menu. The ERD Tool allows you to:

- Design and visualize the database tables and their relationships.
- Add notes to the diagram.
- Auto-align the tables and links for cleaner visualization.
- Save the diagram and open it later to continue working on it.
- Generate ready to run SQL from the database design.
- Generate the database diagram for an existing database.



You can open multiple copies of the ERD tool in individual tabs simultaneously. To close a copy of the ERD tool, click the **X** in the upper-right hand corner of the tab bar.

## Toolbar

The **ERD Tool** toolbar uses context-sensitive icons that provide shortcuts to frequently performed tasks. The option is enabled for the highlighted icon and is disabled for the grayed-out icon.



Hover over an icon on Toolbar to display a tooltip that describes the icon's functionality.

File Options

| Icon      | Behavior  | Shortcut         |
|-----------|---|------------------|
| Open File | Click the <b>Open File</b> icon to load a previously saved diagram.   | Ctrl + O         |
| Save      | Click the <b>Save</b> icon to perform a quick-save of a previously saved diagram, or to save the diagram to a file. | Ctrl + S         |
| Save as   | Click the <b>Save As</b> to open a new browser dialog and specify a new location to save the diagram.               | Ctrl + Shift + S |

Export Options

| Icon           | Behavior   | Shortcut          |
|----------------|--|-------------------|
| Generate SQL   | Click the <b>Generate SQL</b> icon to generate the DDL SQL for the diagram and open a query tool with the generated SQL ready for execution. | Option + Ctrl + S |
| Download image | Click the <b>Download image</b> icon to save the ERD diagram in a image format.  | Option + Ctrl + I |

Note

If `ENABLE_DATA_ACCESS_TOOLS` configuration option is set to False then **Generate SQL** option is disabled.

Editing Options

| Icon            | Behavior  | Shortcut              |
|-----------------|---|-----------------------|
| Add table       | Click this button to add a new table to the diagram. On clicking, this will open a table dialog where you can put the table details.  | Option/Alt + Ctrl + A |
| Edit table      | Click this button to edit a table on the diagram. On clicking, this will open a table dialog where you can change table details. This will enable when a table is selected. | Option/Alt + Ctrl + E |
| Clone table     | Click this button to clone the complete table structure, name it with a auto generated name and put it in the diagram.  | Option/Alt + Ctrl + C |
| Drop table/link | You can drop a table or link using this button. You need to select a table or link and click on this button to drop it.   | Option/Alt + Ctrl + D |

Table Relationship Options

| Icon | Behavior  | Shortcut              |
|------|---|-----------------------|
| 1M   | Click this button to open a one-to-many relationship dialog to add a relationship between the two tables. The selected table becomes the referencing table and will have the <b>many</b> endpoint of the link.          | Option/Alt + Ctrl + O |
| MM   | Click this button to open a many-to-many relationship dialog to add a relationship between the two tables. This option will create a new table based on the selected columns for the two relating tables and link them. | Option/Alt + Ctrl + M |

Utility Options



| Icon          | Behavior   | Shortcut               |
|---------------|--|------------------------|
| Add/Edit note | Click this button to make notes on tables nodes while designing the database.                                | Option/Alt + Ctrl + N  |
| Auto align    | Click this button to auto align all tables and links to make it look more cleaner.                           | Option/Alt + Ctrl + L  |
| Show details  | Click this button to toggle the column details visibility. It allows you to show few or more column details. | Option/Alt + Shift + D |

Zoom Options

| Icon        | Behavior   | Shortcut                 |
|-------------|--|--------------------------|
| Zoom to fit | Click this button to zoom in/out automatically and fit all the tables to the view. | Option/Alt + Shift + F   |
| Zoom in     | Click this button to zoom in the diagram.  | Option/Alt + Shift + "+" |
| Zoom out    | Click this button to zoom out the diagram.   | Option/Alt + Shift + "-" |

Table Dialog

New table

GeneralColumns

Name

users1

Schema

public

Comment

users information table

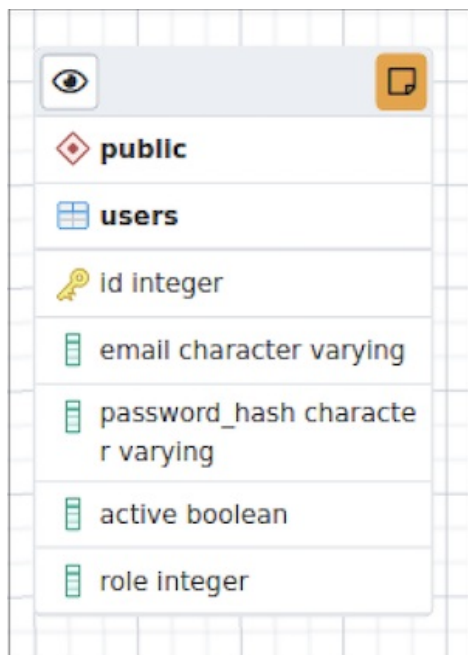
Cancel

OK

The table dialog allows you to:

- Change the table structure details.
- It can be used edit an existing table or add a new one.
- Refer [table dialog](#) for information on different fields.

Table Node



The table node shows table details in a graphical representation:

- The top bar has a **details toggle button** that is used to toggle column details visibility. There is also a **note button** that is visible only if there is some note added. you can click on this button to quickly change the note.
- The first row shows the schema name of the table. Eg. **public** in above image.
- The second row shows the table name. Eg. **users** in above image.
- All other rows below the table name are the columns of the table along with data type. If the column is a primary key then it will have lock key icon eg. id is the primary key in above image. Otherwise, it will have column icon.
- You can click on the node and drag to move on the canvas.
- Upon double click on the table node or by clicking the edit button from the toolbar, the table dialog opens where you can change the table details. Refer [table dialog](#) for information on different fields.

### The One to Many Link Dialog

One to many relation

General

Local Table

(public) roles

Local Column

role\_id

Referenced Table

(public) users

Referenced Column

role

Cancel

OK

The one to many link dialog allows you to:

- Add a foreign key relationship between two tables.
- **Local Table** is the table that references a table and has the **many** end point.
- **Local Column** is the column that references.
- **Referenced Table** is the table that is being referred and has the **one** end point.
- **Referenced Column** is the column that is being referred.

### The Many to Many Link Dialog

Many to many relation

General

Left Table

(public) roles

Left table Column

role\_id

Right Table

(public) user\_detail

Right table Column

id

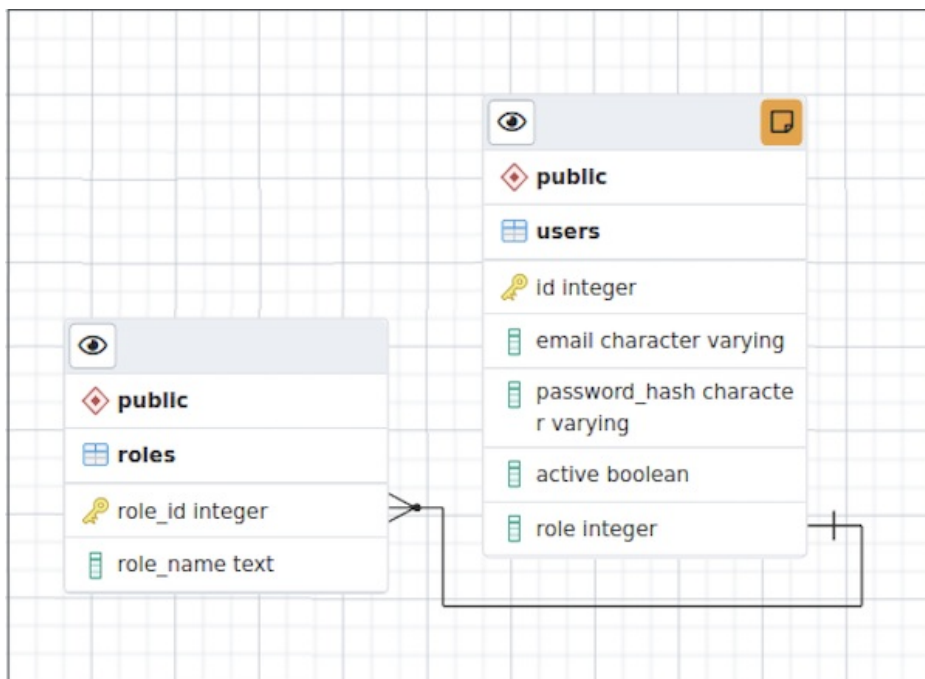
Cancel

OK

The many to many link dialog allows you to:

- Add a many to many relationship between two tables.
- It creates a relationship tables having columns derived from the two tables and link them to the tables.
- **Left Table** is the first table that is to be linked. It will receive the **one** endpoint of the link with the new relation table.
- **Left Column** is the column of the first table, that will always be a primary key.
- **Right Table** is the second table that is to be linked. It will receive the **one** endpoint of the link with the new relation table.
- **Right Column** is the column of the second table, that will always be a primary key.

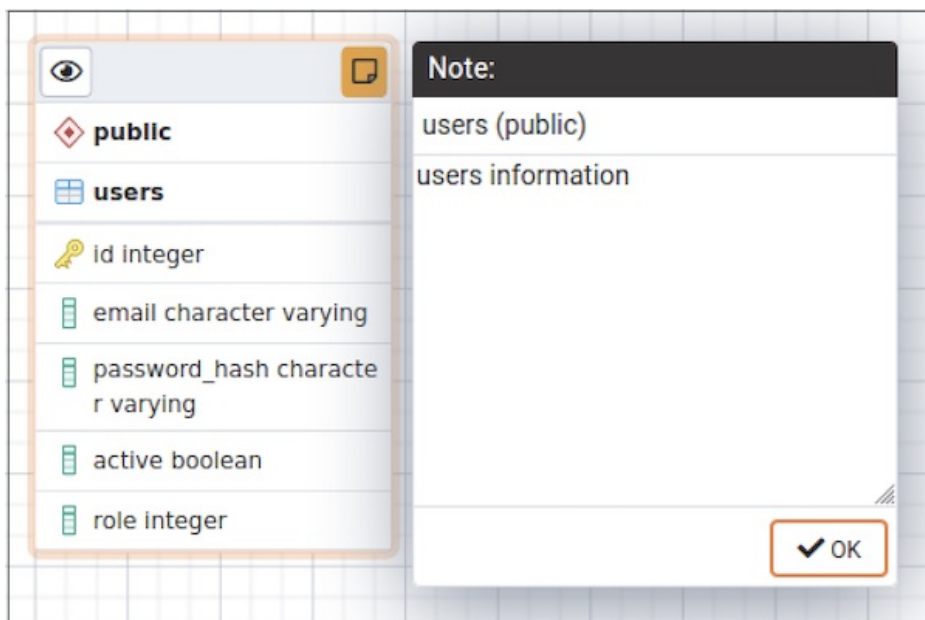
## The Table Link



The table link shows relationship between tables:

- The single line endpoint of the link shows the column that is being referred.
- The three line endpoint of the link shows the column that refers.
- If one of the columns that is being referred or that refers is removed from the table then the link will get dropped.
- You can click on the link and drag to move on the canvas.

### The Table Notes



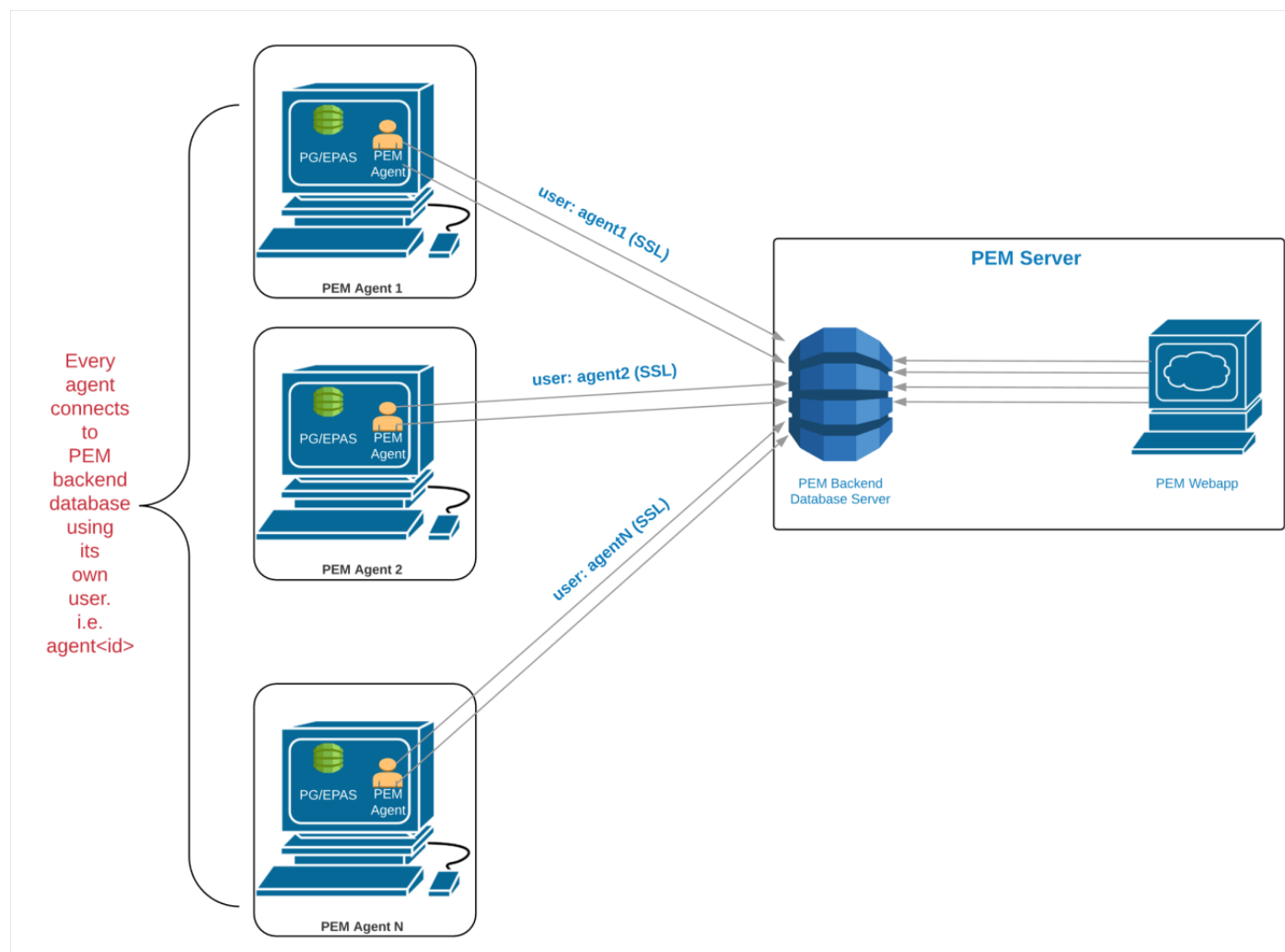
- You can use the notes popup to mark some notes while designing the database.
- You open the pop-up using the toolbar note button.
- If some note is added to a table then it will have notes button on the table node. You can click on the button to check/update notes.

## 34.9 Configuring pgBouncer for use with PEM Agents

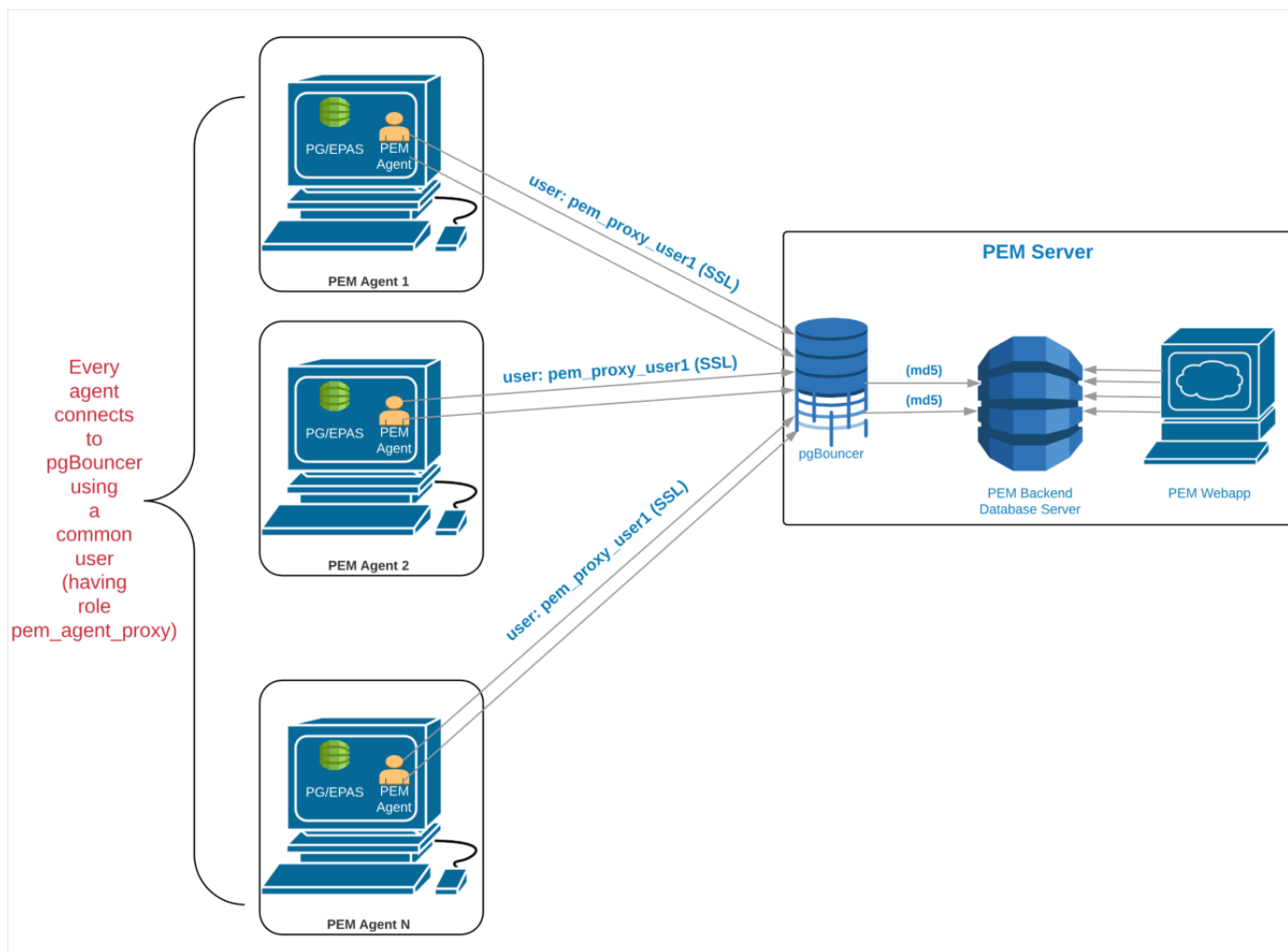
pgBouncer is a lightweight connection pooler for Postgres. You can use pgBouncer to limit the number of connections from the PEM Agent towards the Postgres Enterprise Manager (PEM) server on a non-Windows machine.

### 34.9.1 Connecting PEM to pgBouncer

Each PEM agent connects to the PEM database server using the SSL certificates for each individual user. For example, an agent with ID#1 connects to the PEM database server using the agent1 user.



Prior to PEM version 7.5, the following limitations did not allow use of the connection pooler between the PEM server and PEM agent: \* The PEM agent uses an SSL Certificate to connect the PEM database server. \* It uses an individual user identifier when connecting to the PEM database server. EnterpriseDB has modified the PEM agent to allow the agent to use a common database user (instead of the dedicated agent users) to connect the PEM database server.



We recommend using pgBouncer versions equal to or later than version 1.9.0 as the connection pooler. Since versions 1.9.0 or later support cert authentication; PEM Agents can connect to pgBouncer using SSL certificates.

## 34.9.2 Preparing the PEM Server for pgBouncer Connections

You must configure the PEM database server to accept connections from pgBouncer; the following example demonstrates the steps required to prepare the PEM database server.

1. Create a dedicated user named `pgbouncer` on the PEM database server. For example:

```
pem=# CREATE USER pgbouncer PASSWORD 'ANY_PASSWORD' LOGIN;
CREATE ROLE
```

2. Create a user named `pem_admin1` (a non-super user) with `pem_admin` and `pem_agent_pool` role membership on the PEM database server. For example:

```
pem=# CREATE USER pem_admin1 PASSWORD 'ANY_PASSWORD' LOGIN CREATEROLE;
CREATE ROLE
pem=# GRANT pem_admin, pem_agent_pool TO pem_admin1;
GRANT ROLE
```

3. Grant `CONNECT` privilege to the `pgbouncer` user on the `pem` database. For example:

```
pem=# GRANT CONNECT ON DATABASE pem TO pgbouncer ;GRANT USAGE ON SCHEMA pem TO pgbouncer;
GRANT
```

- Grant USAGE privilege to the pgbouncer user for the pem schema on the pem database. For example:

```
pem=# GRANT USAGE ON SCHEMA pem TO pgbouncer;
GRANT
```

- Grant EXECUTE privilege to the pgbouncer user on the pem.get\_agent\_pool\_auth(text) function in the pem database. For example:

```
pem=# GRANT EXECUTE ON FUNCTION pem.get_agent_pool_auth(text) TO pgbouncer;
GRANT
```

- Use the pem.create\_proxy\_agent\_user(varchar) function to create a user named pem\_agent\_user1 on the PEM database server. The function will create a user with the same name with a random password, and grant `pem_agent` and `pem_agent_pool` roles to the user. This allows pgBouncer to use a proxy user on behalf of the agent. For example:

```
pem=# SELECT pem.create_proxy_agent_user('pem_agent_user1');
create_proxy_agent_user
-----
(1 row)
```

- Add the following entries to the start of the `pg_hba.conf` file of the PEM database server; this will allow pgBouncer user to connect to the pem database using the md5 authentication method. For example:

```
# Allow the PEM agent proxy user (used by
# pgbouncer) to connect the to PEM server using
# md5

local pem pgbouncer,pem_admin1 md5
```

After configuring the PEM server, you should configure pgBouncer.

### 34.9.3 Configuring pgBouncer

You must configure pgBouncer to work with the PEM database server. In our example, we will run pgBouncer as the enterprisedb system user. The following steps outline the process of configuring pgBouncer (version >= 1.9).

- Open a terminal window and navigate into the pgBouncer directory.
- Change the owner of the etc directory for pgBouncer (where pgbouncer.ini resides) to enterprisedb, and change the directory permissions to 0700. For example:

```
$ chown enterprisedb:enterprisedb /etc/edb/pgbouncer1.9
$ chmod 0700 /etc/edb/pgbouncer1.9
```

- Change the contents of the pgbouncer.ini or edb-pgbouncer.ini file as follows:

```
[databases]
;; Change the pool_size according to maximum connections allowed
;; to the PEM database server as required.
;; 'auth_user' will be used for authenticate the db user (proxy
```



```

;; agent user in our case)
pem = port=5444 host=/tmp dbname=pem auth_user=pgbouncer pool_size=80 pool_mode=transaction
* = port=5444 host=/tmp dbname=pem auth_user=pgbouncer pool_size=10
[pgbouncer]
logfile = /var/log/edb/pgbouncer1.9/edb-pgbouncer-1.9.log
pidfile = /var/run/edb/pgbouncer1.9/edb-pgbouncer-1.9.pid
listen_addr = *
;; Agent needs to use this port to connect the pem database now
listen_port = 6432
;; Require to support for the SSL Certificate authentications
;; for PEM Agents
client_tls_sslmode = require
;; These are the root.crt, server.key, server.crt files present
;; in the present under the data directory of the PEM database
;; server, used by the PEM Agents for connections.
client_tls_ca_file = /var/lib/edb/as11/data/root.crt
client_tls_key_file = /var/lib/edb/as11/data/server.key
client_tls_cert_file = /var/lib/edb/as11/data/server.crt
;; Use hba file for client connections
auth_type = hba
;; Authentication file, Reference:
;; https://pgbouncer.github.io/config.html#auth_file
auth_file = /etc/edb/pgbouncer1.9/userlist.txt
;; HBA file
auth_hba_file = /etc/edb/pgbouncer1.9/hba_file
;; Use pem.get_agent_pool_auth(TEXT) function to authenticate
;; the db user (used as a proxy agent user).
auth_query = SELECT * FROM pem.get_agent_pool_auth($1)
;; DB User for administration of the pgbouncer
admin_users = pem_admin1
;; DB User for collecting the statistics of pgbouncer
stats_users = pem_admin1
server_reset_query = DISCARD ALL
;; Change based on the number of agents installed/required
max_client_conn = 500
;; Close server connection if its not been used in this time.
;; Allows to clean unnecessary connections from pool after peak.
server_idle_timeout = 60

```

4. Use the following command to create and update the /etc/edb/pgbouncer1.9/userlist.txt authentication file for pgBouncer:

```

pem=# COPY (
  SELECT 'pgbouncer'::TEXT, 'pgbouncer_password'
  UNION ALL
  SELECT 'pem_admin1'::TEXT, 'pem_admin1_password'
) TO '/etc/edb/pgbouncer1.9/userlist.txt'
  WITH (FORMAT CSV, DELIMITER ' ', FORCE_QUOTE *);
COPY 2

```

NOTE: A super user cannot invoke the PEM authentication query function `pem.get_proxy_auth(text)`. If the `pem_admin` user is a super user, you must add the password to the authentication file, which is `enterprisedb` in the above example.

5. Create an HBA file (/etc/edb/pgbouncer1.9/hba\_file) for pgBouncer that contains the following content:

```

# Use authentication method md5 for the local connections to
# connect pem database & pgbouncer (virtual) database.
local pgbouncer all md5

# Use authentication method md5 for the remote connections to

```

```
# connect to pgbouncer (virtual database) using enterprisedb
# user.
host pgbouncer,pem pem_admin1 0.0.0.0/0 md5
# Use authentication method cert for the TCP/IP connections to
# connect the pem database using pem_agent_user1
hostssl pem pem_agent_user1 0.0.0.0/0 cert
```

6. Change the owner of the HBA file (/etc/edb/pgbouncer1.9/hba\_file) to enterprisedb, and change the directory permissions to 0600. For example:

```
$ chown enterprisedb:enterprisedb /etc/edb/pgbouncer1.9/hba_file
$ chmod 0600 /etc/edb/pgbouncer1.9/hba_file
```

7. Enable the pgBouncer service, and start the service. For example:

```
$ systemctl enable edb-pgbouncer-1.9
Created symlink from /etc/systemd/system/multi-user.target.wants/edb-pgbouncer-1.9.service to
/usr/lib/systemd/system/edb-pgbouncer-1.9.service.
$ systemctl start edb-pgbouncer-1.9
```

### 34.9.4 Configuring the PEM Agent to use pgBouncer

You can use an RPM package to install a PEM Agent; for detailed installation information, please see the [PEM Installation docs](#)

Please note that PEM Agent which is responsible for sending SNMP notifications should not be configured with pgBouncer. For Example - If default PEM Agent installed along with PEM Server is used for SNMP notifications, then it should not be configured with pgBouncer.

#### Configuring a New PEM Agent (installed using an RPM)

After using an RPM package to install the PEM agent, you will need to configure it to work it against a particular PEM database server. Use the following command:

```
$ PGSSLMODE=require PEM_SERVER_PASSWORD=pem_admin1_password /usr/edb/pem/agent/bin/pemworker --register-
agent --pem-server 172.16.254.22 --pem-port 6432 --pem-user pem_admin1 --pem-agent-user pem_agent_user1 --
display-name *Agent_Name*
Postgres Enterprise Manager Agent registered successfully!
```

In above command, the command line argument --pem-agent-user instructs the agent to create an SSL certificate and key pair for the pem\_agent\_user1 database user in /root/.pem directory. For example:

```
/root/.pem/pem_agent_user1.crt
/root/.pem/pem_agent_user1.key
```

They will be used by the PEM agent to connect to the PEM database server as pem\_agent\_user1. It will also create /usr/edb/pem/agent/etc/agent.cfg.

You will find a line mentioning the agent-user to be used in the agent.cfg configuration file. For example:

```
$ cat /usr/edb/pem/agent/etc/agent.cfg
[PEM/agent]
pem_host=172.16.254.22
pem_port=6432
```

```

agent_id=12
agent_user=pem_agent_user1
agent_ssl_key=/root/.pem/pem_agent_user1.key
agent_ssl_cert=/root/.pem/pem_agent_user1.crt
log_level=warning
log_location=/var/log/pem/worker.log
agent_log_location=/var/log/pem/agent.log
long_wait=30
short_wait=10
alert_threads=0
enable_smtp=false
enable_snmp=false
enable_webhook=false
max_webhook_retries=3
allow_server_restart=true
max_connections=0
connect_timeout=-1
connection_lifetime=0
allow_batch_probes=false
heartbeat_connection=false

```

## Configuring an Existing PEM Agent (installed using an RPM)

If you are using an existing PEM agent, you can copy the SSL certificate and key files to the target machine, and reuse the files. You will need to modify the files, adding a new parameter and replacing some parameters in the existing `agent.cfg` file.

Add a line for `agent_user` to be used for the agent. For example:

```
agent_user=pem_agent_user1
```

Update the port to specify the pgBouncer port. For example:

```
pem_port=6432
```

Update the certificate and key path locations. For example:

```

agent_ssl_key=/root/.pem/pem_agent_user1.key
agent_ssl_cert=/root/.pem/pem_agent_user1.crt

```

Please note: as an alternative, you can run the agent self registration, but that will create a new agent id. If you do run the agent self-registration, you must replace the new agent id with existing id, and disable the entry for the new agent id in the `pem.agent` table. For example:

```

pem=# UPDATE pem.agent SET active = false WHERE id = *new_agent_id*;
UPDATE 1

```

Please keep a backup of the existing SSL certificate, key file, and agent configuration file.

## 34.10 pgAgent

pgAgent is a job scheduling agent for Postgres databases, capable of running multi-step batch or shell scripts and SQL tasks on complex schedules.

pgAgent is distributed independently. You can download pgAgent from the [download area](#) of the pgAdmin website.

### 34.10.1 Using pgAgent

pgAgent is a scheduling agent that runs and manages jobs; each job consists of one or more steps and schedules. If two or more jobs are scheduled to execute concurrently, pgAgent will execute the jobs in parallel (each with its own thread).

A step may be a series of SQL statements or an operating system batch/shell script. Each step in a given job is executed when the previous step completes, in alphanumeric order by name. Switches on the **pgAgent Job** dialog (accessed through the **Properties** context menu) allow you to modify a job, enabling or disabling individual steps as needed.

Each job is executed according to one or more schedules. Each time the job or any of its schedules are altered, the next runtime of the job is re-calculated. Each instance of pgAgent periodically polls the database for jobs with the next runtime value in the past. By polling at least once every minute, all jobs will normally start within one minute of the specified start time. If no pgAgent instance is running at the next runtime of a job, it will run as soon as pgAgent is next started, following which it will return to the normal schedule.

When you highlight the name of a defined job in the browser tree control, the **Properties** tab of the main PEM window will display details about the job, and the **Statistics** tab will display details about the job's execution.

#### Security concerns

pgAgent is a very powerful tool, but does have some security considerations that you should be aware of:

**Database password** - **DO NOT** be tempted to include a password in the pgAgent connection string - on Unix systems it may be visible to all users in 'ps' output, and on Windows systems it will be stored in the registry in plain text. Instead, use a libpq `~/.pgpass` file to store the passwords for every database that pgAgent must access. Details of this technique may be found in the [PostgreSQL documentation on .pgpass file](#).

**System/database access** - all jobs run by pgAgent will run with the security privileges of the pgAgent user. SQL steps will run as the user that pgAgent connects to the database as, and batch/shell scripts will run as the operating system user that the pgAgent service or daemon is running under. Because of this, it is essential to maintain control over the users that are able to create and modify jobs. By default, only the user that created the pgAgent database objects will be able to do this - this will normally be the PostgreSQL superuser.

### 34.10.2 Installing pgAgent

pgAgent runs as a daemon on Unix systems, and a service on Windows systems. In most cases it will run on the database server itself - for this reason, pgAgent is not automatically configured when PEM is installed. In some cases however, it may be preferable to run pgAgent on multiple systems, against the same database; individual jobs may be targeted at a particular host, or left for execution by any host. Locking prevents execution of the same instance of a job by multiple hosts.

#### Database setup

Before using PEM to manage pgAgent, you must create the pgAgent extension in the maintenance database registered with PEM. To install pgAgent on a PostgreSQL host, connect to the **postgres** database, and navigate through the **Tools** menu to open the Query tool. For server versions 9.1 or later, and pgAgent 3.4.0 or later, enter the following command in the query window, and click the **Execute** icon:

```
CREATE EXTENSION pgagent;
```

This command will create a number of tables and other objects in a schema called 'pgagent'.

The database must also have the pl/pgsql procedural language installed - use the PostgreSQL `CREATE LANGUAGE` command to install pl/pgsql if necessary. To install pl/pgsql, enter the following command in the query window, and click the `Execute` icon:

```
CREATE LANGUAGE plpgsql;
```

## Daemon installation on Unix

### Note

pgAgent is available in Debian/Ubuntu (DEB) and Redhat/Fedora (RPM) packages for Linux users, as well as source code. See the [pgAdmin Website](#) for more information.

To install the pgAgent daemon on a Unix system, you will normally need to have root privileges to modify the system startup scripts. Modifying system startup scripts is quite system-specific so you should consult your system documentation for further information.

The program itself takes few command line options, most of which are only needed for debugging or specialised configurations:

### Usage:

```
/path/to/pgagent [options] <connect-string>
```

### options:

```
-f run in the foreground (do not detach from the terminal)
-t <poll time interval in seconds (default 10)>
-r <retry period after connection abort in seconds (>=10, default 30)>
-s <log file (messages are logged to STDOUT if not specified)>
-l <logging verbosity (ERROR=0, WARNING=1, DEBUG=2, default 0)>
```

The connection string is a standard PostgreSQL libpq connection string (see the [PostgreSQL documentation on the connection string](#) for further details). For example, the following command line will run pgAgent against a server listening on the localhost, using a database called 'postgres', connecting as the user 'postgres':

```
/path/to/pgagent hostaddr=127.0.0.1 dbname=postgres user=postgres
```

## Service installation on Windows

### Note

pgAgent is available in a pre-built installer if you use [EnterpriseDB's PostgreSQL Installers](#). Use the StackBuilder application to download and install it. If installed in this way, the service will automatically be created and the instructions below can be ignored.

pgAgent can install itself as a service on Windows systems. The command line options available are similar to those on Unix systems, but include an additional parameter to tell the service what to do:

### Usage:

```
pgAgent REMOVE <serviceName>
pgAgent INSTALL <serviceName> [options] <connect-string>
pgAgent DEBUG [options] <connect-string>
```

## options:

```
-u <user or DOMAIN\user>
-p <password>
-d <displayname>
-t <poll time interval in seconds (default 10)>
-r <retry period after connection abort in seconds (>=10, default 30)>
-l <logging verbosity (ERROR=0, WARNING=1, DEBUG=2, default 0)>
```

The service may be quite simply installed from the command line as follows (adjust the path as required):

```
"C:\Program Files\pgAgent\bin\pgAgent" INSTALL pgAgent -u postgres -p secret hostaddr=127.0.0.1
dbname=postgres user=postgres
```

You can then start the service at the command line using `net start pgAgent`, or from the **Services** control panel applet. Any logging output or errors will be reported in the Application event log. The DEBUG mode may be used to run pgAgent from a command prompt. When run this way, log messages will output to the command window.

### 34.10.3 Creating a pgAgent Job

pgAgent is a scheduling agent that runs and manages jobs; each job consists of steps and schedules.

To create or manage a job, use the Browser tree control to browse to the server on which the pgAgent database objects were created. The tree control will display a **pgAgent Jobs** node, under which currently defined jobs are displayed. To add a new job, right click on the **pgAgent Jobs** node, and select **Create pgAgent Job...** from the context menu.

When the pgAgent dialog opens, use the tabs on the **pgAgent Job** dialog to define the steps and schedule that make up a pgAgent job.

**Create - pgAgent Job**

General Steps Schedules SQL

Name: eom\_batch\_process

Enabled?: ☒ Yes

Job class: Data Import  
Please select a class to categorize the job. This option will not affect the way the job runs.

Host agent:   
Enter the hostname of a machine running pgAgent if you wish to ensure only that machine will run this job. Leave blank if any host may run the job.

Comment:

Buttons: [i] [?] [Cancel] [Reset] [Save]

Use the fields on the **General** tab to provide general information about a job:

- Provide a name for the job in the **Name** field.

- Move the **Enabled** switch to the **Yes** position to enable a job, or **No** to disable a job.
- Use the **Job Class** drop-down to select a class (for job categorization).
- Use the **Host Agent** field to specify the name of a machine that is running pgAgent to indicate that only that machine may execute the job. Leave the field blank to specify that any machine may perform the job.

**Note:** It is not always obvious what value to specify for the Host Agent in order to target a job step to a specific machine. With pgAgent running on the required machines and connected to the scheduler database, you can use the following query to view the hostnames as reported by each agent:



```
SELECT jagstation FROM pgagent.pga_jobagent
```



Use the hostname exactly as reported by the query in the Host Agent field.

- Use the **Comment** field to store notes about the job.

Create - pgAgent Job

GeneralStepsSchedulesSQL

|   | Name | Enabled?        | Kind           | Connection type  | On error        |
|---|------|-----------------|----------------|------------------|-----------------|
|   | ID   | <div>True</div> | <div>SQL</div> | <div>Local</div> | <div>Fail</div> |

Cancel

Reset

Save

Use the **Steps** tab to define and manage the steps that the job will perform. Click the Add icon (+) to add a new step; then click the compose icon (located at the left side of the header) to open the step definition dialog:

**Create - pgAgent Job**

General Steps Schedules SQL

|                                     | Name | Enabled? | Kind | Connection type | On error |
|-------------------------------------|------|----------|------|-----------------|----------|
| <input checked="" type="checkbox"/> | ID   | True     | SQL  | Local           | Fail     |

**General** Code

Name: ID

Enabled?: Yes

Kind: SQL

Connection type: Local  
Select **Local** if the job step will execute on the local database server, or **Remote** to specify a remote database server.

Database: postgres  
Please select the database on which the job step will run.

Connection string:  
Please specify the connection string for the remote database server. Each parameter setting is in the form keyword = value. Spaces around the equal sign are optional. To write an empty value, or a value containing spaces, surround it with single quotes, e.g., keyword = 'a value'. Single quotes and backslashes within the value must be escaped with a backslash, i.e., ' and \.  
For more information, please see the documentation on [libpq connection strings](#).

On error: Fail

Comment:

Cancel Reset Save

Use fields on the step definition dialog to define the step:

- Provide a name for the step in the **Name** field; please note that steps will be performed in alphanumeric order by name.
- Use the **Enabled** switch to include the step when executing the job (**True**) or to disable the step (**False**).
- Use the **Kind** switch to indicate if the job step invokes SQL code (**SQL**) or a batch script (**Batch**).
  - If you select **SQL**, use the **Code** tab to provide SQL code for the step.
  - If you select **Batch**, use the **Code** tab to provide the batch script that will be executed during the step.

#### Note

The fields **Connection type**, **Database** and **Connection string** are only applicable when **SQL** is selected because **Batch** cannot be run on remote servers.

- Use the **Connection type** switch to indicate if the step is performed on a local server (**Local**) or on a remote host (**Remote**). If you specify a remote connection should be used for the step, the **Connection string** field will be enabled, and you must provide a libpq-style connection string.
- Use the **Database** drop-down to select the database on which the job step will be performed.
- Use the **Connection string** field to specify a libpq-style connection string to the remote server on which the step will be performed. For more information about writing a connection string, please see the [PostgreSQL documentation](#).
- Use the **On error** drop-down to specify the behavior of pgAgent if it encounters an error while executing the step. Select from:
  - **Fail** - Stop the job if you encounter an error while processing this step.



- **Success** - Mark the step as completing successfully, and continue.
  - **Ignore** - Ignore the error, and continue.
- Use the **Comment** field to provide a comment about the step.

**Create - pgAgent Job**

General Steps Schedules SQL

|                                     | Name | Enabled?                            | Kind | Connection type | On error |
|-------------------------------------|------|-------------------------------------|------|-----------------|----------|
| <input checked="" type="checkbox"/> | ID   | <input checked="" type="checkbox"/> | SQL  | Local           | Fail     |

General Code

```

1 BEGIN
2 END;

```

Cancel Reset Save



Use the context-sensitive field on the step definition dialog's **Code** tab to provide the SQL code or batch script that will be executed during the step:



- If the step invokes SQL code, provide one or more SQL statements in the **SQL query** field.
- If the step performs a batch script, provide the script in the **Script** field. If you are running on a Windows server, standard batch file syntax must be used. When running on a Linux server, any shell script may be used, provided that a suitable interpreter is specified on the first line (e.g. `#!/bin/sh`).

When you've provided all of the information required by the step, click the compose icon to close the step definition dialog. Click the add icon (+) to add each additional step, or select the **Schedules** tab to define the job schedule.

**Create - pgAgent Job**

General Steps Schedules SQL



|   | Name          | Enabled?                                 | Start                      | End                        |
|---|---------------|--|----------------------------|----------------------------|
|   | test_schedule | <input checked="" type="checkbox"/> True | 2020-04-27 12:08:06 +05:30 | 2020-04-27 12:08:08 +05:30 |

  Cancel Reset Save

Click the Add icon (+) to add a schedule for the job; then click the compose icon (located at the left side of the header) to open the schedule definition dialog:

**Create - pgAgent Job**


General Steps Schedules SQL


|   | Name          | Enabled?                                 | Start                      | End                        |
|---|---------------|--|----------------------------|----------------------------|
|   | test_schedule | <input checked="" type="checkbox"/> True | 2020-04-27 12:08:06 +05:30 | 2020-04-27 12:08:08 +05:30 |

General Repeat Exceptions



Name: test\_schedule

Enabled?: ☒ Yes

Start: 2020-04-27 12:08:06 +05:30 

End: 2020-04-27 12:08:08 +05:30 

Comment:

  Cancel Reset Save

Use the fields on the schedule definition tab to specify the days and times at which the job will execute.

- Provide a name for the schedule in the **Name** field.
- Use the **Enabled** switch to indicate that pgAgent should use the schedule ( **Yes** ) or to disable the schedule ( **No** ).
- Use the calendar selector in the **Start** field to specify the starting date and time for the schedule.

- Use the calendar selector in the **End** field to specify the ending date and time for the schedule.
- Use the **Comment** field to provide a comment about the schedule.

Select the **Repeat** tab to define the days on which the schedule will execute.

**Create - Agent Job**

General Steps **Schedules** Notifications SQL

| Name | Enabled? | Start                   | End                     |
|------|----------|-------------------------|-------------------------|
| test | True     | 2022-01-10 15:25 +05:30 | 2022-01-10 15:25 +05:30 |

General **Repeat** Exceptions

Schedules are specified using a **cron-style** format.  
 For each selected time or date element, the schedule will execute.  
 e.g. To execute at 5 minutes past every hour, simply select '05' in the Minutes list box.  
 Values from more than one field may be specified in order to further control the schedule.  
 e.g. To execute at 12:05 and 14:05 every Monday and Thursday, you would click minute 05, hours 12 and 14, and weekdays Monday and Thursday.  
 For additional flexibility, the Month Days check list includes an extra Last Day option. This matches the last day of the month, whether it happens to

**Days**

Week Days: Sunday

Month Days: 1st

Months: January

**Times**

Hours: 01

Minutes: 00

Timezone: Asia/Calcutta (IST)

Select the timezone to be used in the next job run.

Cancel Reset Save

Use the fields on the **Repeat** tab to specify the details about the schedule in a cron-style format. The job will execute on each date or time element selected on the **Repeat** tab.

Click within a field to open a list of valid values for that field; click on a specific value to add that value to the list of selected values for the field. To clear the values from a field, click the X located at the right-side of the field.

Use the fields within the **Days** box to specify the days on which the job will execute:

- Use the **Week Days** field to select the days on which the job will execute.
- Use the **Month Days** field to select the numeric days on which the job will execute. Specify the **Last Day** to indicate that the job should be performed on the last day of the month, irregardless of the date.
- Use the **Months** field to select the months in which the job will execute.

Use the fields within the **Times** box to specify the times at which the job will execute:

- Use the **Hours** field to select the hour at which the job will execute.
- Use the **Minutes** field to select the minute at which the job will execute.
- Use the **Timezone** drop-down to select the timezone to be used in the next job run.

Select the **Exceptions** tab to specify any days on which the schedule will **not** execute.

Create - pgAgent Job

GeneralStepsSchedulesSQL

|                                     | Name          | Enabled?                            | Start                      | End                        |
|-------------------------------------|---------------|-------------------------------------|----------------------------|----------------------------|
| <input checked="" type="checkbox"/> | test_schedule | <input checked="" type="checkbox"/> | 2020-04-27 12:08:06 +05:30 | 2020-04-27 12:08:08 +05:30 |

GeneralRepeatExceptions

|                          | Date       | Time  |
|--------------------------|------------|-------|
| <input type="checkbox"/> | 2020-04-27 | 12:09 |

?

Cancel

Reset

Save

Use the fields on the **Exceptions** tab to specify days on which you wish the job to not execute; for example, you may wish for jobs to not execute on national holidays.

Click the Add icon (+) to add a row to the exception table, then:

- Click within the **Date** column to open a calendar selector, and select a date on which the job will not execute. Specify **<Any>** in the **Date** column to indicate that the job should not execute on any day at the time selected.
- Click within the **Time** column to open a time selector, and specify a time on which the job will not execute. Specify **<Any>** in the **Time** column to indicate that the job should not execute at any time on the day selected.

When you've finished defining the schedule, you can use the **SQL** tab to review the code that will create or modify your job.

Create - pgAgent Job

GeneralStepsSchedulesSQL

```
1 DO $$
2 DECLARE
3     jid integer;
4     scid integer;
5 BEGIN
6 -- Creating a new job
7 INSERT INTO pgagent.pga_job(
8     jobjclid, jobname, jobdesc, jobhostagent, jobenabled
9 ) VALUES (
10    1::integer, 'eom_batch_process'::text, ''::text, ''::text, true
11 ) RETURNING jobid INTO jid;
12
13 -- Steps
14 -- Inserting a step (jobid: NULL)
15 INSERT INTO pgagent.pga_jobstep (
16     jstjobid, jstname, jstenabled, jstkind,
17     jstconnstr, jstdbname, jstonerror,
18     jstcode, jstdesc
19 ) VALUES (
20     jid, 'ID'::text, true, 's'::character(1),
21     ''::text, 'postgres'::name, 'f'::character(1),
22     'BEGIN
23 END;'::text, ''::text
24 ) ;
25
26 -- Schedules
27 -- Inserting a schedule
28 INSERT INTO pgagent.pga_schedule(
29     jscjobid, jscname, jscdesc, jscenabled,
30     jscstart, jscend,     jscminutes, jschours, jscweekdays, jscmonthdays, jscmonths
31 ) VALUES (
32     jid, 'test schedule'::text, ''::text, true,
```

i?

CancelResetSave

Click the **Save** button to save the job definition, or **Cancel** to exit the job without saving. Use the **Reset** button to remove your unsaved entries from the dialog.

After saving a job, the job will be listed under the **pgAgent Jobs** node of the browser tree control of the server on which it was defined. The **Properties** tab in the main PEM window will display a high-level overview of the selected job, and the **Statistics** tab will show the details of each run of the job.

Dashboard

Properties

SQL

Statistics

Dependencies

Dependents

Monitoring

i

Edit

▼ General

Name

eom\_batch\_process

ID

1

Enabled?

Yes

Job class

Routine Maintenance

Host agent

Created

2020-04-26 23:39:56.264597-07

Changed

2020-04-26 23:39:56.264597-07

Next run

Last run

Last result

Unknown

Running at

Not running currently.

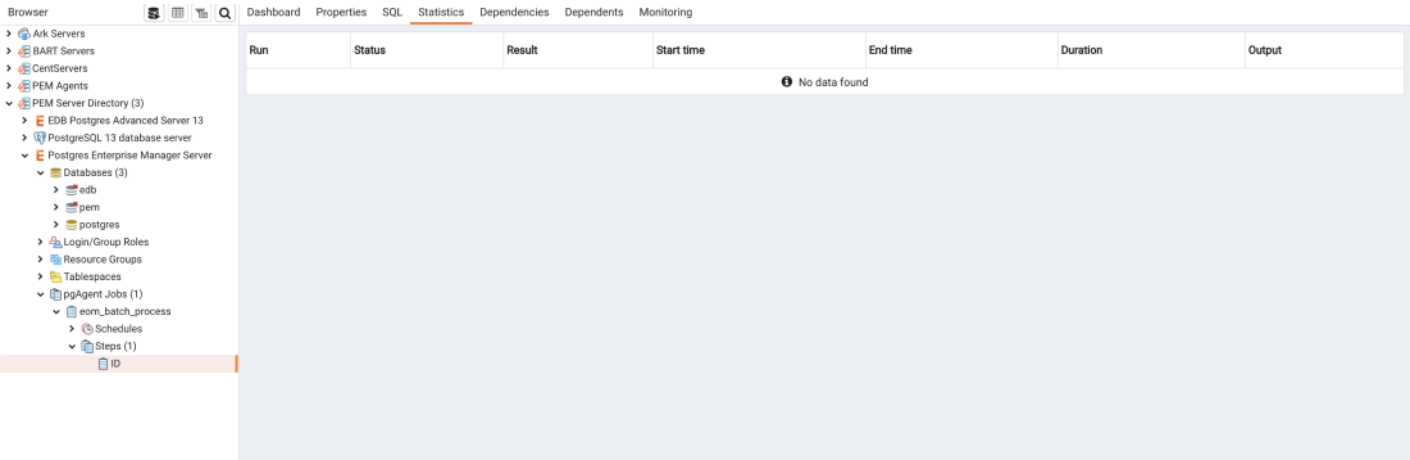
Comment

To modify an existing job or to review detailed information about a job, right-click on a job name, and select **Properties** from the context menu.

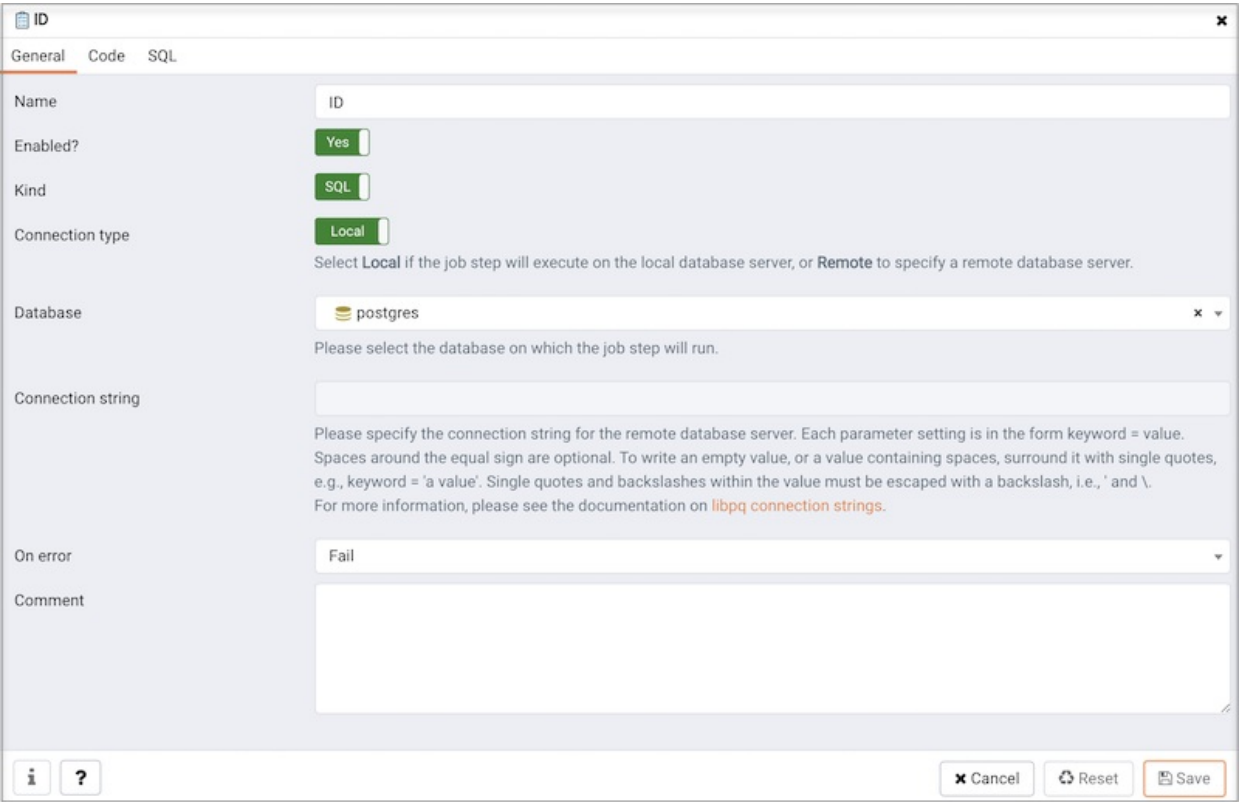
### 34.10.4 pgAgent Steps

Each Job consists of a number of steps, each of which may be an SQL script, or an operating system batch/shell script. Each step in a given job is run in turn, in alphanumeric name order.

Steps may be added to a job through the job properties dialogue, or added as a sub-object. The **Properties** tab of the main PEM client window will display details of the selected step, and the **Statistics** tab will display details of each run of the step, including and output or errors from the script.



Each step consists of the details shown on the screenshot below, most of which are self-explanatory. If **Kind** is set to SQL, then it goes without saying that a database against which to run the script must be selected. If set to **Batch**, the database/connection string should be left blank. The **On Error** option controls how failure of this step will affect the status of the overall job.



The **Definition** tab contains a single text box into which the step script should be entered. For SQL steps, this should be a series of one or more SQL statements. For batch jobs, when running on a Windows server, standard batch file syntax must be used, and when running on a *nix* server, *any shell script may be used, provided that a suitable interpreter is specified on the first line (e.g.#!/bin/sh\*)*.

34.10.5 pgAgent Schedules

Each Job is executed according to one or more schedules. Each time the job or any of its schedules are altered, the next runtime of the job is re-calculated. Each instance of pgAgent periodically polls the database for jobs with the next runtime value in the past. By polling at least once every minute, all jobs will normally start within one minute of the specified start time. If no pgAgent instance is running at the next runtime of a job, it will run as soon as pgAgent is next started, following which it will return to the normal schedule.

Schedules may be added to a job through the job properties dialogue, or added as a sub-object. The **Properties** tab of the main PEM client window will display details of the selected schedule.

Dashboard

Properties

SQL

Statistics

Dependencies

Dependents

Monitoring

i

Edit

General

ID

20

Name

test

Enabled?

Yes

Start

2022-01-10 15:25 +05:30

End

2022-01-10 15:25 +05:30

Week days

Sunday

Month days

1st

Months

January

Hours

01

Minutes

00


Timezone

Asia/Calcutta (IST)

Exceptions

Comment


Each schedule consists of the basic details such as a name, whether or not it is enable and a comment. In addition, a start date and time is specified (before which the schedule has no effect), and optionally an end date and time (after which the schedule has no effect).



**test\_schedule**
✕

General
Repeat
Exceptions
SQL



Name
test\_schedule


Enabled?
☒ Yes


Start
2020-04-27 12:08:06 +05:30


End
2020-04-27 12:08:08 +05:30


Comment

✕ Cancel
 Reset

 Save

Schedules are specified using a cron-style format. For each selected time or date element, the schedule will execute. For example, to execute at 5 minutes past every hour, simply tick '5' in the **Minutes** list box. Values from more than one field may be specified in order to further control the schedule. For example, to execute at 12:05 and 14:05 every Monday and Thursday, you would tick minute 5, hours 12 and 14, and weekdays Monday and Thursday. For additional flexibility, the **Month Days** check list includes an extra **Last Day** option. This matches the last day of the month, whether it happens to be the 28th, 29th, 30th or 31st. Use **Timezone** drop-down to select the timezone to be used in the next job run.



test

General
Repeat
Exceptions

Schedules are specified using a **cron-style** format.

For each selected time or date element, the schedule will execute.

e.g. To execute at 5 minutes past every hour, simply select '05' in the Minutes list box.

Values from more than one field may be specified in order to further control the schedule.

e.g. To execute at 12:05 and 14:05 every Monday and Thursday, you would click minute 05, hours 12 and 14, and weekdays Monday and Thursday.

For additional flexibility, the Month Days check list includes an extra Last Day option. This matches the last day of the month, whether it happens to be the 28th, 29th, 30th or 31st.

Days

Week Days
Month Days
Months

Times

Hours
Minutes
Timezone

Select the timezone to be used in the next job run.

i
?
Cancel
Reset
Save

On occasion it may be desirable to specify an exception for a schedule - for example, you may not want a schedule to fire on a particular national holiday. To achieve this, each schedule may have a list of date and/or time exceptions attached to it. If a schedule lands on an exception, that instance will be skipped, and the following occurrence will become the next runtime.

### 34.11.1 Licence

### 34.11.2 The MIT Kerberos Licence

PostgreSQL Enterprise Manager uses PostgreSQL's libpq library which may be linked with MIT Kerberos Libraries on some distributions. The MIT Kerberos licence is included below:

### Kerberos Copyright

This software is being provided to you, the LICENSEE, by the Massachusetts Institute of Technology (M.I.T.) under the following license. By obtaining, using and/or copying this software, you agree that you have read, understood, and will comply with these terms and conditions:

Permission to use, copy, modify and distribute this software and its documentation for any purpose and without fee or royalty is hereby granted, provided that you agree to comply with the following copyright notice and statements, including the disclaimer, and that the same appear on ALL copies of the software and documentation, including modifications that you make for internal use or for distribution:

Copyright 1992-2004 by the Massachusetts Institute of Technology. All rights reserved.

THIS SOFTWARE IS PROVIDED "AS IS", AND M.I.T. MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. By way of example, but not limitation, M.I.T. MAKES NO REPRESENTATIONS OR WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE LICENSED SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

The name of the Massachusetts Institute of Technology or M.I.T. may NOT be used in advertising or publicity pertaining to distribution of the software. Title to copyright in this software and any associated documentation shall at all times remain with M.I.T., and USER agrees to preserve same.

Project Athena, Athena, Athena MUSE, Discuss, Hesiod, Kerberos, Moira, OLC, X Window System, and Zephyr are trademarks of the Massachusetts Institute of Technology (MIT). No commercial use of these trademarks may be made without prior written permission of MIT.

### 34.11.3 The OpenSSL Licence

Postgres Enterprise Manager uses code from the OpenSSL project to provide support for SSL encrypted connections. The OpenSSL licence is included below:

**Copyright (c) 1998-2011 The OpenSSL Project. All rights reserved.**

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (<http://www.openssl.org/>)"
- The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact [openssl-core@openssl.org](mailto:openssl-core@openssl.org).
- Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project.
- Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>)"

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes cryptographic software written by Eric Young ([ey@cryptsoft.com](mailto:ey@cryptsoft.com)). This product includes software written by Tim Hudson ([tjh@cryptsoft.com](mailto:tjh@cryptsoft.com)).

### 34.11.4 The SNMP++ Licence

Postgres Enterprise Manager uses code from the SNMP++ project to send snmp v1/v2 notifications. The SNMP++ licence is included below:

Copyright (c) 2001-2010 Jochen Katz, Frank Fock

This software is based on SNMP++2.6 from Hewlett Packard:

Copyright (c) 1996 Hewlett-Packard Company

ATTENTION: USE OF THIS SOFTWARE IS SUBJECT TO THE FOLLOWING TERMS. Permission to use, copy, modify, distribute and/or sell this software and/or its documentation is hereby granted without fee. User agrees to display the above copyright notice and this license notice in all copies of the software and any documentation of the software. User agrees to assume all liability for the use of the software; Hewlett-Packard and Jochen Katz make no representations about the suitability of this software for any purpose. It is provided "AS-IS" without warranty of any kind, either express or implied. User hereby grants a royalty-free license to any and all derivatives based upon this software code base.

### 34.11.5 The jquery table sort Licence

TABLESORT.JS Copyright, Andy Croxall ([mitya@mitya.co.uk](mailto:mitya@mitya.co.uk)) For documentation and demo see <http://mitya.co.uk/scripts/Animated-table-sort-REGEXP-friendly-111>

USAGE This script may be used, distributed and modified freely but this header must remain in tact. For usage info and demo, including info on args and params, see [www.mitya.co.uk/scripts](http://www.mitya.co.uk/scripts)

## 34.12 PEM Worker Usage Guide

The PEM Worker performs tasks on behalf of the PEM Agent. Please note that all the pemworker command-line options must be run as root user on Linux platforms and as an administrator user on Windows platforms.

The following command-line options are supported by the PEM Worker:

On Linux:

```

$ ./pemworker
Postgres Enterprise Manager Agent
Version: 8.0.0
(Revision: REL-7_15_0-5-685-g415f794, Built on: Dec 1 2020 13:27:02)
Schema Supported: 202010081

./pemworker [options]
./pemworker --register-agent [register-options]
./pemworker --register-server [register-server-options]
./pemworker --unregister-server [unregister-server-options]
./pemworker --unregister-agent [unregister-agent-options]
./pemworker --add-webhook [add-webhook-options]
./pemworker --remove-webhook [remove-webhook-options]
./pemworker --register-barman [barman-registration-options]
./pemworker --update-barman [barman-update-options]
```

```
./pemworker --unregister-barman [barman-unregistration-options]
```

#### options:

```
-v/--version (Show the version & the supported schema and, then exit.)
-c <filepath> (Path to the configuration file.)
-o <option name>=<option value> (Override the configuration file options.)
-f Run in the foreground (do not detach from the terminal.)
-t <short_wait> (Default: 10, Poll time interval (in seconds).)
-r <long_wait> (Default: 30, Retry period (in seconds) after connection aborted.)
-s <log file> (Path to the worker log file. Messages are logged to STDOUT if not specified.)
-l <logging verbosity> (Default: ERROR, Possible Values: ERROR, WARNING, DEBUG1, DEBUG2)
--allow_server_restart <true|false> (Default: true, Allow to restart the monitored server.)
--allow_batch_probes <true|false> (Default: false, Allow to run the batch probes on this agent.)
--batch-script-user <os-user> (Default: <none>, Name of the operating system user to run the batch
script that is defined as a batch probe and a step in jobs.)
--enable-heartbeat-connection (Default: false, Agent to use dedicated connection to update the
heartbeat)
--enable-smtp <true|false> (Default: false, Allow the PEM agent to send the email on behalf of the PEM
server.)
--enable-snmp <true|false> (Default: false, Allow the PEM agent to send the SNMP traps on behalf of the
PEM server.)
--enable-webhook <true|false> (Default: false, Allow the PEM agent to call webhooks on behalf of the
PEM server.)
--max-webhook-retries (Default: 3, Retry webhooks for specified times on failure.)
```

#### register-options:

```
--pem-server <hostname|address> (Host name/address of the PEM backend database server.)
--pem-port <port number> (Default: 5432, Port of the PEM backend database server.)
--pem-user <username> (Database user, having superuser privileges, of the PEM backend database server.)
--pem-agent-user <username> (Default: agent<agent_id>, Agent user to connect the PEM server backend
database server)
--cert-path <path> (Default: "~/pem", Path where SSL certificates are saved.)
--config-dir <path> (Default: "<pemworker path>/../etc", Directory path where configuration file can be
found.)
--display-name <name> (Default: system hostname, Display name for the agent.)
--force-registration (forcefully registers the agent to PEM server with arguments provided.)
--group <name> (Specify the group name where registered agent will be displayed.)
--team <database-role> (Specify the name of the database role, on the PEM backend database server, that
should have access to the monitored database server.)
--owner <database-user> <Specify the name of the database user, on the PEM backend database server, who
will own the agent.)
--allow_server_restart <true|false> (Default: true, Allow to restart the monitored server.)
--allow_batch_probes <true|false> (Default: false, Allow to run the batch probes on this agent.)
--batch-script-user <os-user> (Default: <none>, Name of the operating system user to run the batch
script that is defined as a batch probe and a step in jobs.)
--enable-heartbeat-connection (Default: false, Agent to use dedicated connection to update the
heartbeat)
--enable-smtp <true|false> (Default: false, Allow the PEM agent to send the email on behalf of the PEM
server.)
--enable-snmp <true|false> (Default: false, Allow the PEM agent to send the SNMP traps on behalf of the
PEM server.)
--enable-webhook <true|false> (Default: false, Allow the PEM agent to call webhooks on behalf of the
PEM server.)
--max-webhook-retries (Default: 3, Retry webhooks for specified times on failure.)
-o <option name>=<option value> (Override the configuration file options.)
```

#### NOTE:

Set environment variable PEM\_SERVER\_PASSWORD to provide the password for the PEM server to allow the pemworker to register itself.

#### register-server-options:

```

--pem-user <username> (Database user, a member of 'pem_admin' role, of the PEM backend database
server.)
--display-name <name> (Default: system hostname, Display name for the monitored database server.)
--server-addr <hostname|address> (Host name/address of the database server to register with PEM
server.)
--server-port <port> (Port of the database server.)
--server-database <name>(Postgres server database.)
--server-user <name>(Database server user name.)
--server-service-name <service-name> (Database server service name.)
--efm-cluster-name <name> (EFM Cluster name.)
--efm-install-path <path> (EFM installation path.)
--config-dir <Directory path of agent configuration file (Default: "<pemworker path>/../etc")
--remote-monitoring <yes|no> (Default: 'no', Allow remote monitoring.)
--group <name> (Specify the group name where registered server will be displayed.)
--team <database-role> (Specify the name of the database role, on the PEM backend database server, that
should have access to the monitored database server.)
--owner <name> (Specify the name of the database user who will own the monitored server.)
--asb-host-name <hostname> (Host name that the agent will use to connect to the database server. If not
provided then '--server-addr' value will be used.)
--asb-host-port <port> (Port number that agent will use to connect to database server. If not provided
then '--server-port' value will be used.)
--asb-host-db <database> (Database name that agent will use to connect to database server. If not
provided then '--server-database' value will be used.)
--asb-host-user <database-user> (Database username that agent will use to connect to database server.
If not provided then '--server-user' value will be used.)
--asb-ssl-mode <ssl-mode> (Default: 'prefer', SSL mode that agent will use to connect to database
server.)

```

#### NOTE:

Set environment variable PEM\_SERVER\_PASSWORD to provide the password for the PEM server to allow the pemworker to connect as a PEM admin user. Set PEM\_MONITORED\_SERVER\_PASSWORD to provide the password of the database server being registered and monitored by pemagent.

#### unregister-server-options:

```

--pem-user <username> (Database user, a member of 'pem_admin' role, of the PEM backend database
server.)
--server-addr (Database server address to unregister from PEM server.)
--server-port (Port of database server.)
--config-dir <Directory path of agent configuration file (Default: "<pemworker path>/../etc")

```

#### unregister-agent-options:

```

--pem-user <username> (Database user, a member of 'pem_admin' role, of the PEM backend database
server.)
--config-dir <Directory path of agent configuration file (Default: "<pemworker path>/../etc")

```

#### add-webhook-options:

```

--webhook-name (Name of the webhook endpoint.)
--webhook-ssl-key (The complete path of the client key file to connect to the webhook server.)
--webhook-ssl-crt (The complete path of the client certificate file to connect to the webhook server.)
--webhook-ssl-ca-crt (The complete path to the webhook SSL CA certificate file.)
--webhook-ssl-crl (The complete path of the CRL file to validate revoked webhook server certificate.)
--webhook-allow-insecure-ssl (Default: false, Allow webhooks to call with insecure flag.)
--config-dir <Directory path of agent configuration file (Default: "<pemworker path>/../etc")

```

#### remove-webhook-options:

```

--webhook-name (Name of the webhook endpoint.)
--config-dir <Directory path of agent configuration file (Default: "<pemworker path>/../etc")

```

#### barman-registration-options:

```

--api-url <url> (URL of the BARMAN API 'pg-backup-api')

```

```

--description <name> (Description to show on the UI 'User interface' for the BARMAN API.)
--probe-execution-frequency <interval> (Default: 30, Probe the BARMAN API 'pg-backup-api' at regular
interval 'in seconds' and fetch the metrics.)
--heartbeat-interval <interval> (Default: 10, Ping the BARMAN API 'pg-backup-api' 'status' API at a
regular interval 'in seconds' for checking its availability.)
--ssl-crt <certificate_file> (SSL certificate file for the BARMAN API.)
--ssl-key <key_file> (Private SSL key for the BARMAN API.)
--ssl-ca-cert <ca_file> (CA certificate to verify peer against the BARMAN API.)
--team <database-role> (Specify the name of the database group role, on the PEM backend database
server, that should have access to this BARMAN API Server.)
--owner <database-user> (Specify the name of the database user, on the PEM backend database server, who
will own the BARMAN API Server.)
--config-file/-c <config_file> (Path to the agent configuration file.)

```

#### barman-update-options:

```

--id <barman-id> (ID for the existing BARMAN API 'pg-backup-api')
--api-url <url> (URL of the BARMAN API 'pg-backup-api')
--probe-execution-frequency <interval> (Default: 30, Probe the BARMAN API 'pg-backup-api' at regular
interval 'in seconds' and fetch the metrics.)
--heartbeat-interval <interval> (Default: 10, Ping the BARMAN API 'pg-backup-api' 'status' API at a
regular interval 'in seconds' for checking its availability.)
--ssl-crt <certificate_file> (SSL certificate file for the BARMAN API.)
--ssl-key <key_file> (Private SSL key for the BARMAN API.)
--ssl-ca-cert <ca_file> (CA certificate to verify peer against the BARMAN API.)
--config-file/-c <config_file> (Path to the agent configuration file.)

```

#### barman-unregistration-options:

```

--id <barman-id> (ID for the existing BARMAN API, registered with the PEM Server.'pg-backup-api')
--config-file/-c <config_file> (Path to the agent configuration file.)

```

#### NOTE:

Set environment variable PEM\_SERVER\_PASSWORD to provide the password for the PEM server to allow the pemworker to connect as a PEM admin user.

## On Windows:

```

```shell-session
C:\Program Files\edb\pem\agent\bin> pemworker.exe --help
Postgres Enterprise Manager Worker
Version: 8.0.0

```

```

pemworker.exe DEBUG [options]
pemworker.exe REGISTER [register-options]
pemworker.exe REGISTER-SERVER [register-server-options]
pemworker.exe UNREGISTER-SERVER [unregister-server-options]
pemworker.exe UNREGISTER-AGENT [unregister-agent-options]
pemworker.exe ADD-WEBHOOK [--add-webhook-options]
pemworker.exe REMOVE-WEBHOOK [--remove-webhook-options]

```

#### options:

```

-v/--version (Show the version & the supported schema and, then exit.)
-c <filepath> (Path to the configuration file.)
-o <option name>=<option value> (Override the configuration file options.)
-t <short_wait> (Default: 10, Poll time interval (in seconds).)
-r <long_wait> (Default: 30, Retry period (in seconds) after connection aborted.)
-l <logging verbosity> (Default: ERROR, Possible Values: ERROR, WARNING, DEBUG1, DEBUG2)
--allow_server_restart <true|false> (Default: true, Allow to restart the monitored server.)
--allow_batch_probes <true|false> (Default: false, Allow to run the batch probes on this agent.)

```

```
--enable-heartbeat-connection (Default: false, Agent to use dedicated connection to update the heartbeat)
--enable-smtp <true|false> (Default: false, Allow the PEM agent to send the email on behalf of the PEM
server.)
--enable-snmp <true|false> (Default: false, Allow the PEM agent to send the SNMP traps on behalf of the
PEM server.)
--enable-webhook <true|false> (Default: false, Allow the PEM agent to call webhooks on behalf of the PEM
server.)
--max-webhook-retries (Default: 3, Retry webhooks for specified times on failure.)
```

#### register-options:

```
--pem-server <hostname|address> (Host name/address of the PEM backend database server.)
--pem-port <port number> (Default: 5432, Port of the PEM backend database server.)
--pem-user <username> (Database user, having superuser privileges, of the PEM backend database server.)
--pem-agent-user <username> (Default: agent<agent_id>, Agent user to connect the PEM server backend
database server)
--cert-path <path> (Default: "%APPDATA%\pem", Path where SSL certificates are saved.)
--display-name <name> (Default: system hostname, Display name for the agent.)
--force-registration (forcefully registers the agent to PEM server with arguments provided.)
--group <name> (Specify the group name where registered agent will be displayed.)
--team <database-role> (Specify the name of the database role, on the PEM backend database server, that
should have access to the monitored database server.)
--owner <database-user> <Specify the name of the database user, on the PEM backend database server, who
will own the agent.)
--allow_server_restart <true|false> (Default: true, Allow to restart the monitored server.)
--allow-batch-probes <true|false> (Default: false, Allow to run the batch probes on this agent.)
--enable-heartbeat-connection (Default: false, Agent to use dedicated connection to update the heartbeat)
--enable-smtp <true|false> (Default: false, Allow the PEM agent to send the email on behalf of the PEM
server.)
--enable-snmp <true|false> (Default: false, Allow the PEM agent to send the SNMP traps on behalf of the
PEM server.)
--enable-webhook <true|false> (Default: false, Allow the PEM agent to call webhooks on behalf of the PEM
server.)
--max-webhook-retries (Default: 3, Retry webhooks for specified times on failure.)
-o <option name>=<option value> (Override the configuration file options.)
```

#### NOTE:

Set environment variable PEM\_SERVER\_PASSWORD to provide the password for the PEM server to allow the pemworker to register itself.

#### register-server-options:

```
--pem-user <username> (Database user, a member of 'pem_admin' role, of the PEM backend database server.)
--display-name <name> (Default: system hostname, Display name for the monitored database server.)
--server-addr <hostname|address> (Host name/address of the database server to register with PEM server.)
--server-port <port> (Port of the database server.)
--server-database <name>(Postgres server database.)
--server-user <name>(Database server user name.)
--server-service-name <service-name> (Database server service name.)
--remote-monitoring <true|false> (Allow remote monitoring.)
--group <name> (Specify the group name where registered server will be displayed.)
--team <database-role> (Specify the name of the database role, on the PEM backend database server, that
should have access to the monitored database server.)
--owner <name> (Specify the name of the database user who will own the monitored server.)
--asb-host-name <hostname> (Host name that the agent will use to connect to the database server. If not
provided then '--server-addr' value will be used.)
--asb-host-port <port> (Port number that agent will use to connect to database server. If not provided
then '--server-port' value will be used.)
--asb-host-db <database> (Database name that agent will use to connect to database server. If not
provided then '--server-database' value will be used.)
--asb-host-user <database-user> (Database username that agent will use to connect to database server. If
not provided then '--server-user' value will be used.)
--asb-ssl-mode <ssl-mode> (Default: 'prefer', SSL mode that agent will use to connect to database
```



```
server.)
```

**NOTE:**

Set environment variable PEM\_SERVER\_PASSWORD to provide the password for the PEM server to allow the pemworker to connect as a PEM admin user. Set PEM\_MONITORED\_SERVER\_PASSWORD to provide the password of the database server being registered and monitored by pemagent.

**unregister-server-options:**

```
--pem-user <username> (Database user, a member of 'pem_admin' role, of the PEM backend database server.)
--server-addr (Database server address to unregister from PEM server.)
--server-port (Port of database server.)
--config-dir <Directory path of agent configuration file (Default: "<pemworker path>/../etc")
```

**unregister-agent-options:**

```
--pem-user <username> (Database user, a member of 'pem_admin' role, of the PEM backend database server.)
```

**add-webhook-options:**

```
--webhook-name (Name of the webhook endpoint.)
--webhook-ssl-key (The complete path of the client key file to connect to the webhook server.)
--webhook-ssl-crt (The complete path of the client certificate file to connect to the webhook server.)
--webhook-ssl-ca-crt (The complete path to the webhook SSL CA certificate file.)
--webhook-ssl-crl (The complete path of the CRL file to validate revoked webhook server certificate.)
--webhook-allow-insecure-ssl (Default: false, Allow webhooks to call with insecure flag.)
```

**remove-webhook-options:**

```
--webhook-name (Name of the webhook endpoint.)
```

**NOTE:**

Set environment variable PEM\_SERVER\_PASSWORD to provide the password for the PEM server to allow the pemworker to connect as a PEM admin user.

```
...
```