EDB Postgres™ Enterprise Manager
Getting Started Guide
Version 7.6

January 9, 2019

EDB Postgres Enterprise Manager Getting Started Guide
by EnterpriseDB® Corporation

Table of Contents

1 Introduction

This document provides an introduction to Postgres Enterprise Manager™ (PEM). The guide will acquaint you with the basics of the toolset, and help you be successful in your database management activities. The guide is broken up into the following core sections and categories:

- **Postgres Enterprise Manager Overview** – Chapter 2 provides information about PEM functionality, components, architecture, and supported platforms.

- **Using the PEM Web Interface** – Chapter 3 provides an overview of PEM's web interface. The web interface is installed with the PEM server, and can be used from your browser of choice.

- **Registering a Server** - Chapter 4 highlights the different tools available to assist with server registration.

- **General database administration** – Chapter 5 will assist you in performing general database administration tasks, and SQL query and stored procedure development.

- **Server Management** - Chapter 6 provides information about controlling the PEM server, and managing server authentication.

- **PEM Agent Management** - Chapter 7 provides information about configuring and managing a PEM agent.

- **Replacing SSL Certificates** - Chapter 8 provides information about managing SSL certificates.

This guide is not a comprehensive resource; rather, it is meant to serve as an aid to bring you up to speed with the basics of how to use the product. For more information about using PEM features, please see the online help made available through the PEM web interface, or in the PEM guides.

This document uses *Postgres* to mean either the PostgreSQL or EDB Postgres Advanced Server database.

1.1 What's New

The following feature has been added to create Postgres Enterprise Manager 7.6:

- PEM 7.6 adds capability to analyze the database performance for Advanced Server instances by monitoring the wait events. For more information, see the PEM Enterprise Features Guide, available at:

  https://www.enterprisedb.com/resources/product-documentation

1.2 Typographical Conventions Used in this Guide

Certain typographical conventions are used in this manual to clarify the meaning and usage of various commands, statements, programs, examples, etc. This section provides a summary of these conventions.

In the following descriptions a *term* refers to any word or group of words that are language keywords, user-supplied values, literals, etc. A term's exact meaning depends upon the context in which it is used.

- *Italic font* introduces a new term, typically, in the sentence that defines it for the first time.
- `Fixed-width (mono-spaced) font` is used for terms that must be given literally such as SQL commands, specific table and column names used in the examples, programming language keywords, etc. For example, `SELECT * FROM emp;`
- *`Italic fixed-width font`* is used for terms for which the user must substitute values in actual usage. For example, `DELETE FROM `*`table_name`*`;`
- A vertical pipe | denotes a choice between the terms on either side of the pipe. A vertical pipe is used to separate two or more alternative terms within square brackets (optional choices) or braces (one mandatory choice).
- Square brackets [ ] denote that one or none of the enclosed term(s) may be substituted. For example, `[ a | b ]`, means choose one of "a" or "b" or neither of the two.
- Braces {} denote that exactly one of the enclosed alternatives must be specified. For example, `{ a | b }`, means exactly one of "a" or "b" must be specified.
- Ellipses ... denote that the proceeding term may be repeated. For example, `[ a | b ] ...` means that you may have the sequence, "b a a b a".

2 Postgres Enterprise Manager - Overview

Postgres Enterprise Manager (PEM) is an enterprise management tool designed to assist database administrators, system architects, and performance analysts in administering, monitoring, and tuning PostgreSQL and EnterpriseDB Advanced Server database servers. PEM is architected to manage and monitor anywhere from a handful, to hundreds of servers from a single console, allowing complete and remote control over all aspects of your databases.

2.1 Why Postgres Enterprise Manager?

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 keep you up to date on the up/down/performance status of all your servers in an at-a-glance fashion.

- **Distributed Architecture**. PEM is architected in a way that maximizes its 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 via a graphical interface. Server startup and shutdown, configuration management, storage and security control, object creation, performance management, and more can be handled 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 and provide detailed statistics back to easily navigated performance pages within the interface.

- **Proactive Alert Management**. PEM ships out-of-the-box with the ability to create performance thresholds for each key metric (e.g. memory, storage, etc.) that are monitored around-the-clock. Any threshold violation results in an alert being sent to a centralized dashboard that communicates the nature of the problem

and what actions are necessary to prevent the situation from jeopardizing the overall performance of the server.

- **Simplified Capacity Planning**. All key performance-related statistics are automatically collected and retained for a specified period of time in PEM's repository. The Capacity Manager utility allows you to select various statistics and perform trend analysis over time to understand things such as peak load periods, storage consumption trends, and much more. A forecasting mechanism in the tool allows you to also forecast resource usage in the future and plan/budget accordingly.

- **Audit Manager.** The Audit Manager configures audit logging on Advanced Server instances. Activities such as connections to a database, disconnections from a database, and the SQL statements run against a database can be logged. The `Audit Log` dashboard can then be used 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 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 either be done in an ad-hoc or scheduled manner. Captured SQL statements can then be filtered so you can easily identify and tune poorly running SQL statements. SQL statements can also be fed into an Index Advisor on Advanced Server that analyzes each statement and makes recommendations on new indexes that should be created 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, security setup, and much more are examined. Any deviations from recommended best practices are reported back to you, along with an explanation of each particular issue, and expert help on what to do about making things right.

- **Streaming Replication Configuration and Monitoring.** The Streaming Replication wizard simplifies the process of adding new servers to a Postgres streaming replication scenario or configuring existing servers to create a replication scenario. After configuring the replication scenario, you can monitor the scenario on the Streaming Replication dashboard or use options on the PEM client to promote a standby node to the master 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 that allows you to provide connection information for a secure connection.

- **Wide Platform Support**. PEM supports most major Linux and Windows platforms.

2.2 General Architecture

PEM is composed of three primary components (see Figure 2.1):

The PEM Server

> The PEM server provides the functionality at the core of Postgres Enterprise Manager. The server is responsible for:

> - Performing administrative functions.
> - Processing information received from agents.
> - Maintaining information in its repository.

The PEM Agent

> The PEM agent is responsible for performing tasks on each managed machine and collecting statistics for the database server and operating system.

The PEM Web Interface

> Distributed with the PEM server, the PEM web interface allows you to connect to the server with your choice of browser to manage and monitor your Postgres servers.
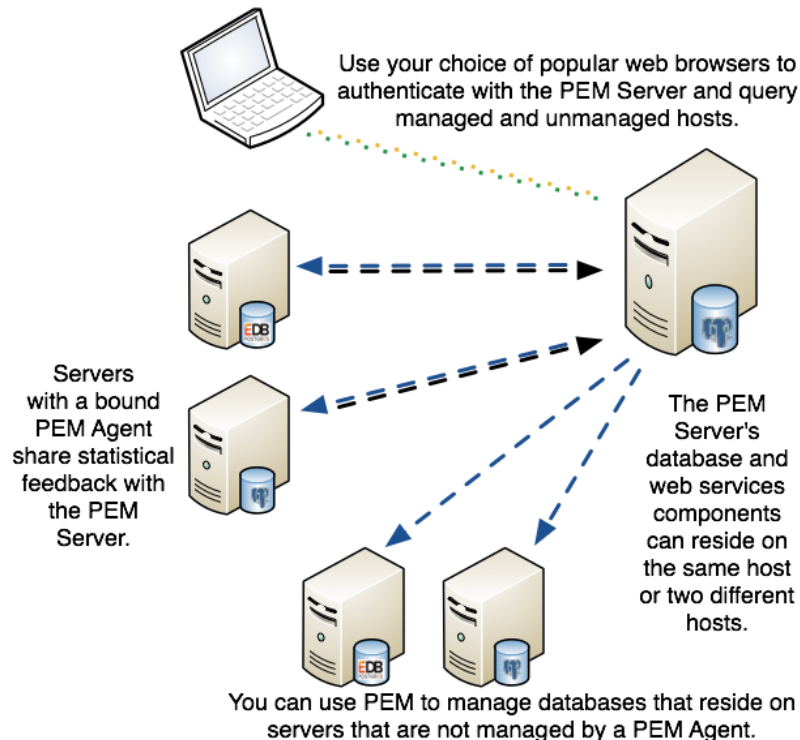

Figure 2.1 - The Postgres Enterprise Manager general architecture.

2.3 Supported Versions and Platforms

The PEM server and agent use Advanced Server or PostgreSQL backing databases (version 9.3 and higher). PEM can manage and monitor Advanced Server and PostgreSQL database versions 9.3 and higher.

The PEM server and agent are supported on:

64-bit Linux:

- CentOS 6.x and 7.x
- Red Hat Enterprise Linux 6.x and 7.x
- OEL 6.x and 7.x
- Ubuntu 16.04 LTS and 14.04 LTS
- Debian 7.x and 8.x
- SLES 12

64-bit Windows:

- Windows Server 2016
- Windows Server 2012 R2

The PEM agent is also supported on Linux ppc64le using RHEL / CentOS 7.x as a PEM server host.

RPM packages for the PEM agent are available for RHEL 6.x and 7.x.

The PEM client is supported on the following browsers:

- Google Chrome
- Mozilla Firefox
- Internet Explorer
- Apple Safari

SQL Profiler is supported on Advanced Server and EnterpriseDB distributions of PostgreSQL version 9.3 and above.

2.4 Installing PEM - Overview

For detailed instructions about installing PEM, please consult the PEM Installation Guide, available at:

The basic steps involved in the PEM installation process are:

1. Install the PEM server components. The PEM server software and backend database (named `pem`) may reside on the same host as the supporting httpd server, or may reside on a separate host.

    The PEM server installer installs a PEM agent and the PEM client on the host of the PEM server.

2. Register each additional physical or virtual machine that you would like to manage with PEM. For convenience, PEM 7 supports remote monitoring (an agent is not required to reside on the same host as the server that it monitors).

    Please note that a remote agent cannot retrieve all of the information available from a monitored server; you may wish to install an agent on each server host.

3. Install the SQL Profiler component into each Postgres instance on which you want to perform SQL capture and analysis. The SQL Profiler installer will prompt you for the location of your Postgres installation, and place the required software into that directory. The SQL Profiler plugin is already installed on Advanced Server instances, and requires only configuration to enable profiling.

3 Using the PEM Web Interface

The PEM web interface is installed with the PEM server. When 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`

Where `ip_address_of_PEM_host` specifies the IP address of the host of the PEM server. The `Postgres Enterprise Manager Login` window (shown in Figure 3.1) opens:
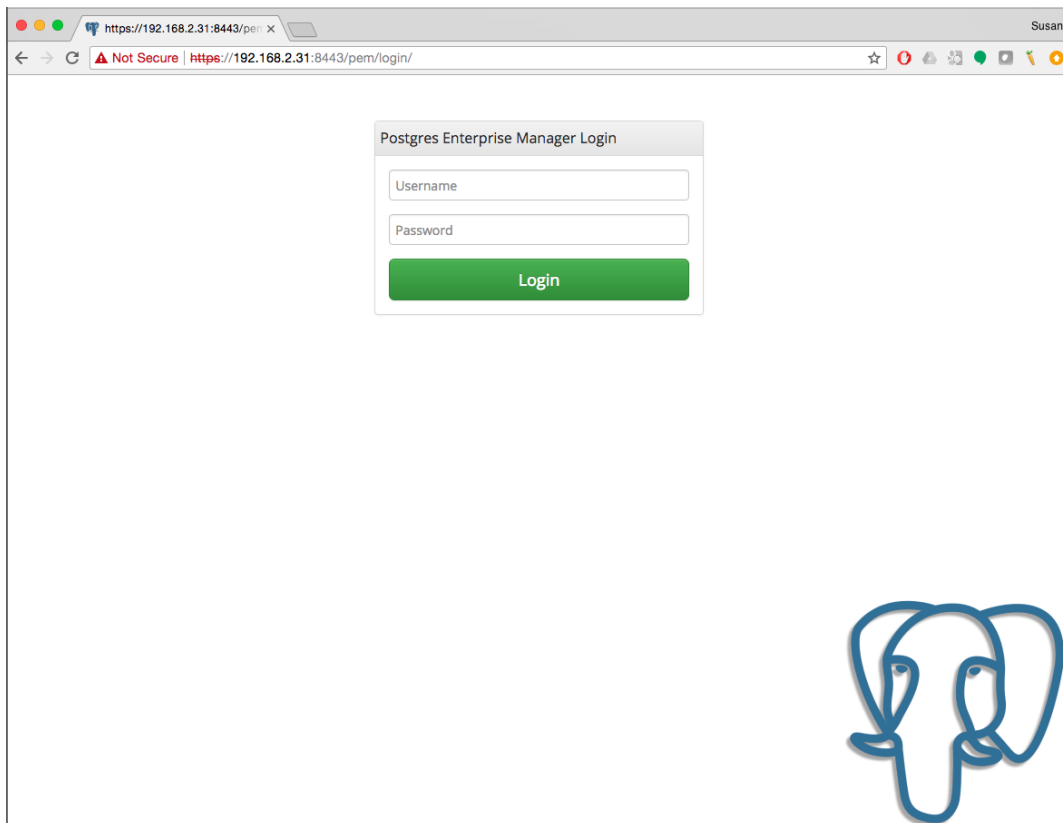

Figure 3.1 - The PEM Login page.

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 will be the name provided when installing the PEM server.

- Provide the password associated with the user in the `Password` field.

After providing your credentials, click `Login` to connect to PEM.

The PEM web interface opens, displaying the `Global Overview Dashboard` (see Figure 3.2).
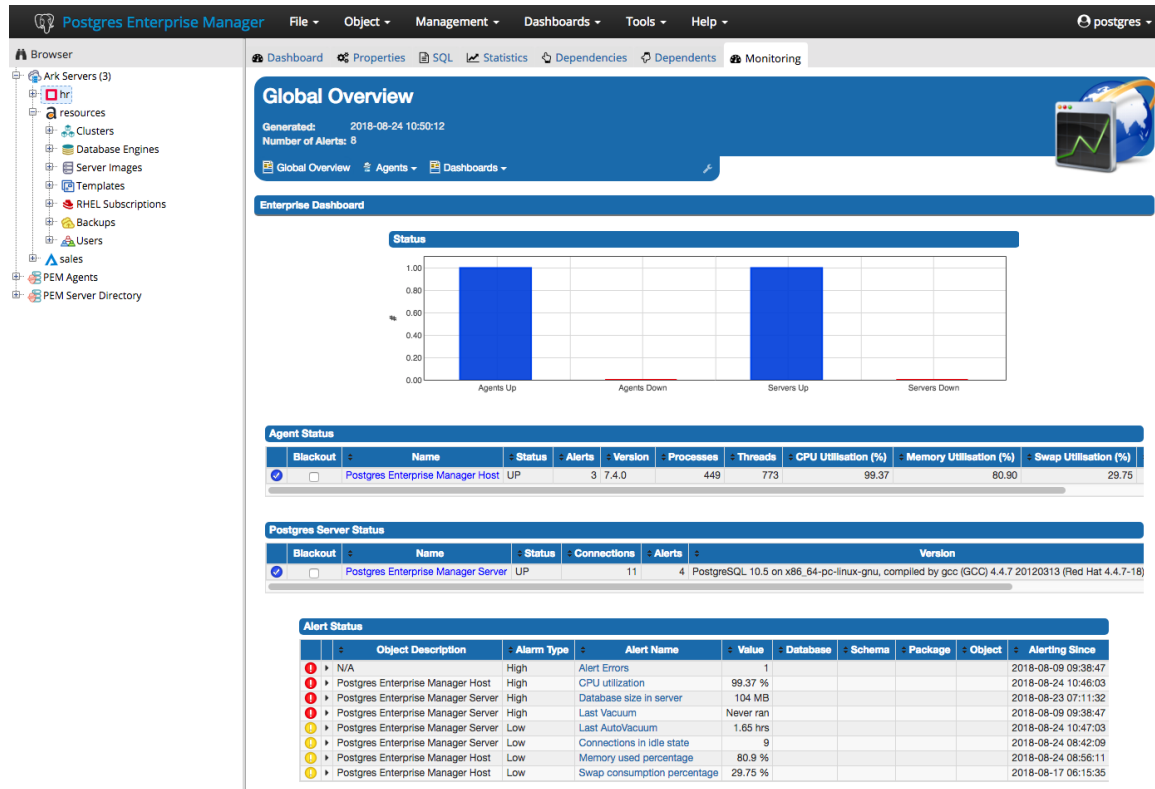

Figure 3.2 - The Global Overview dashboard, displayed in the client.

The `Browser` pane displays a tree control that provides access to information about 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.

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.

3.1 The PEM Toolbar

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.
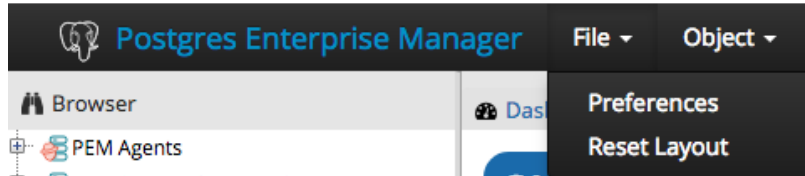
The File Menu

Figure 3.3 – The File Menu

Use the `File` menu (see Figure 3.3) to access the following options:

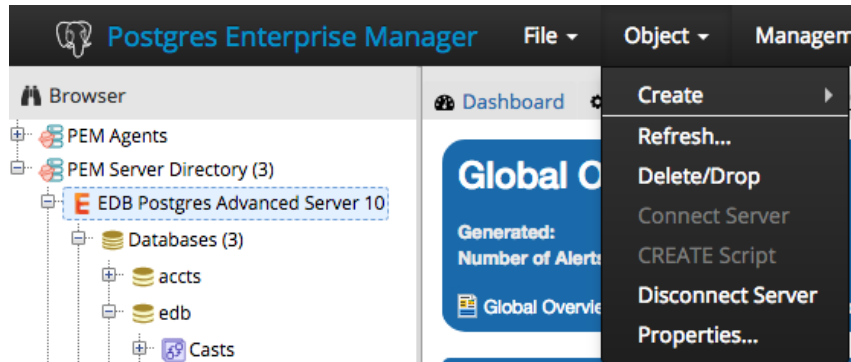| Menu Option	Action
Preferences	Click to open the Preferences dialog to customize your PEM client settings.
Reset Layout	If you have modified the workspace, click to restore the default layout.

The Object Menu

Figure 3.4 – The Object Menu.

The `Object` menu (see Figure 3.4) is context-sensitive. Use the `Object` menu to access the following options:

Menu Option	Action
Connect Server...	Click to open the Connect to Server dialog to establish a connection with a server.
Create	Click *Create* to access a context menu that provides context-sensitive selections.
CREATE Script	Click to open the Query tool to edit or view the selected script.
Delete/Drop	Click to delete the currently selected object from the server.
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.
Hide…	Click to hide the currently selected group; to view hidden groups, enable the Show hidden groups option in Preferences.
Properties...	Click to review or modify the currently selected object's properties.
Refresh...	Click to refresh the currently selected object.
Trigger(s)	Click to *Disable* or *Enable* 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 Data	Click to access a context menu that provides several options for viewing data.

The Management Menu

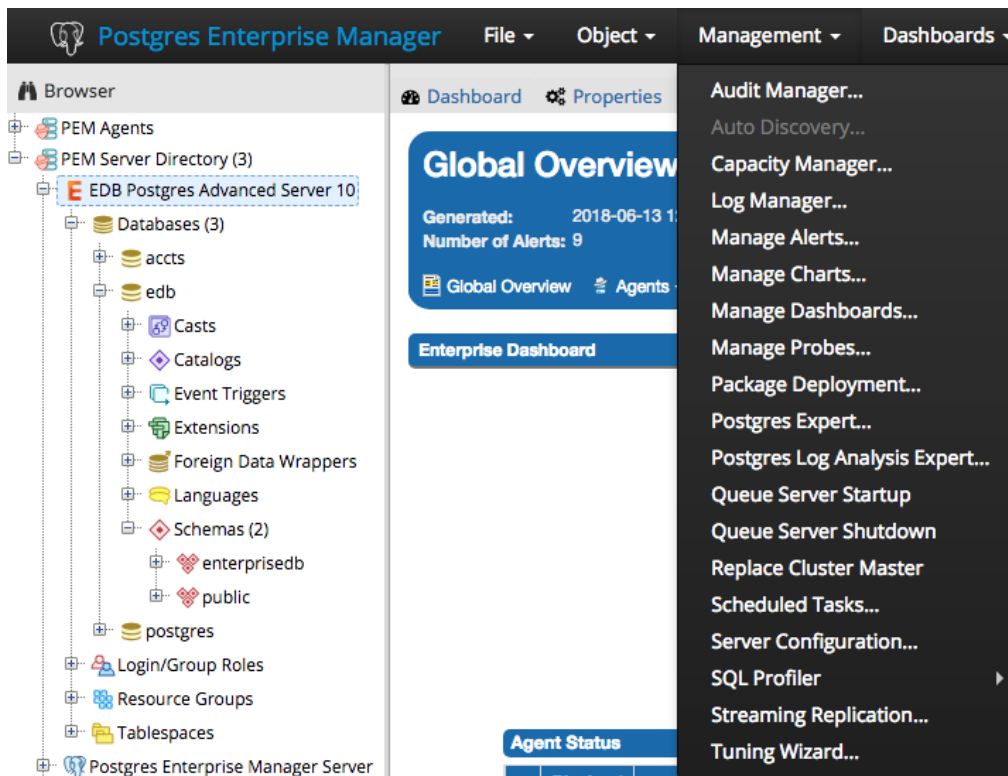*Figure 3.5 – The Management Menu.*

Use the Management menu (see Figure 3.5) to access the following PEM features:

Menu Option	Action
Audit Manager...	Click to open the Audit Manager and configure auditing on your monitored servers.
Auto Discovery...	Click to open the Auto Discovery dialog to instruct a PEM agent to locate and bind monitored database servers.
Capacity Manager...	Click to open the Capacity Manager dialog and analyze historical or project future resource usage.

| Log Manager...	Click to open the Log Manager dialog and configure log collection for a server.
Manage Alerts...	Click to access the Manage Alerts tab and create or modify alerting behavior.
Manage Charts...	Click to open the Manage Charts tab to create or modify PEM charts.
Manage Dashboards...	Click to open the Manage Dashboards dialog to VACUUM, ANALYZE, REINDEX, or CLUSTER.
Manage Probes...	Click to open the Manage Probes dialog to VACUUM, ANALYZE, REINDEX, or CLUSTER.
Package Deployment...	Click to open the Package Deployment wizard and install or update packages.
Postgres Expert...	Click to open the Postgres Expert wizard and perform a static analysis of your servers and databases.
Postgres Log Analysis Expert...	Click to access the Postgres Log Analysis Expert dialog analyze log file contents for usage trends.
Queue Server Startup…	Click to instruct the PEM server to start the currently selected server.
Queue Server Shutdown	Click to instruct the PEM server to shut down the currently selected server.
Replace Cluster Master	Click to replace the current master node of your replication scenario.
Scheduled Tasks	Click to open the Scheduled Tasks tab and review tasks that are pending or recently completed.
Server Configuration...	Click to open the Server Configuration dialog and update your PEM server configuration settings.
SQL Profiler...	Click to open the SQL Profiler dialog to capture statistical information about SQL query performance.
Streaming Replication...	Click to access the Streaming Replication dialog configure a streaming replication scenario.
Tuning Wizard...	Click to open the Tuning Wizard dialog to generate a set of tuning recommendations for your server.

The Dashboards Menu
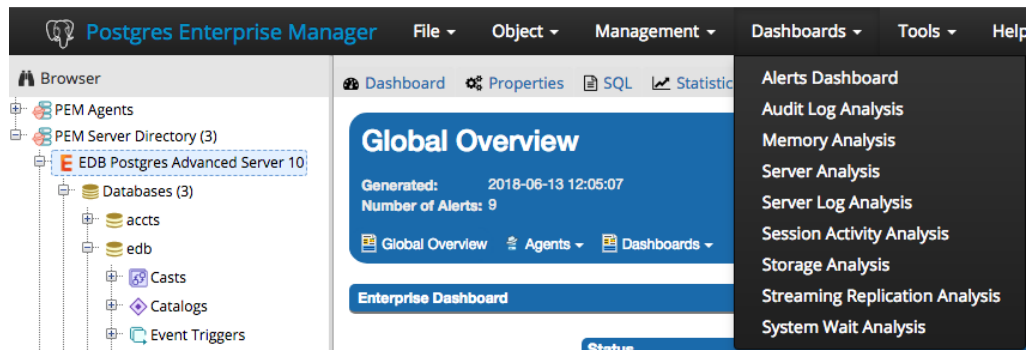
Figure 3.6 – The Dashboards menu.

Use the context-sensitive Dashboards menu (see Figure 3.6) to access dashboards:

Option	Action
Alerts Dashboard	Click to open the Alerts Dashboard for the selected node.
Audit Log Analysis	Click to open the Audit Log Analysis Dashboard for the selected node
Database Analysis	Click to open the Database Analysis Dashboard for the selected node.

|	
I/O Analysis	Click to open the I/O Analysis Dashboard for the selected node.
Memory Analysis	Click to open the Memory Analysis Dashboard for the selected node
Object Activity Analysis	Click to open the Object Activity Analysis Dashboard for the selected node.
Operating System Analysis	Click to open the Operating System Analysis Dashboard for the selected node.
Probe Log Analysis	Click to open the Probe Log Analysis Dashboard for the selected node.
Server Analysis	Click to open the Server Analysis Dashboard for the selected node.
Server Log Analysis	Click to open the Server Log Analysis Dashboard for the selected node.
Session Activity Analysis	Click to open the Session Activity Analysis Dashboard for the selected node.
Storage Analysis	Click to open the Storage Analysis Dashboard for the selected node.
Streaming Replication Analysis	Click to open the Streaming Replication Analysis Dashboard for the selected node.
System Wait Analysis	Click to open the System Wait Analysis Dashboard for the selected node.

The Tools Menu
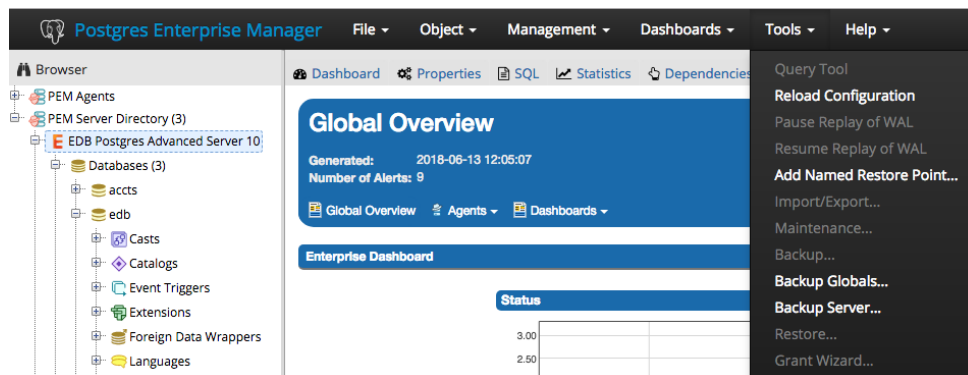
Figure 3.7 – The Tools menu.

Use the options on the `Tools` menu (see Figure 3.7) to access the following features:

Option	Action
Add named restore point	Click to open the dialog to take a point-in-time snapshot of the current server state.
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.
Grant Wizard...	Click to access the Grant Wizard tool.
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.
`Pause replay of WAL`	Click to pause the replay of the WAL log.
`Query tool`	Click to open the Query tool for the currently selected object.
`Reload Configuration...`	Click to update configuration files without restarting the server.
`Restore...`	Click to access the Restore dialog to restore database files from a backup.
`Resume replay of WAL`	Click to resume the replay of the WAL log.

The Help Menu
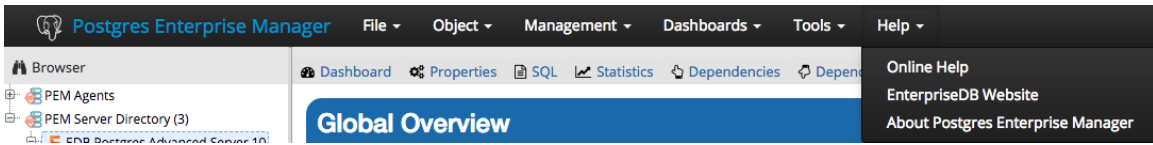
Figure 3.8 – The Help menu.

Use the options on the `Help` menu (see Figure 3.8) to access online help documents or to review information about the PEM installation:

Option	Action
`Online Help`	Click to open documentation for Postres Enterprise Manager.
`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.

3.2 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 (see Figure 3.9).
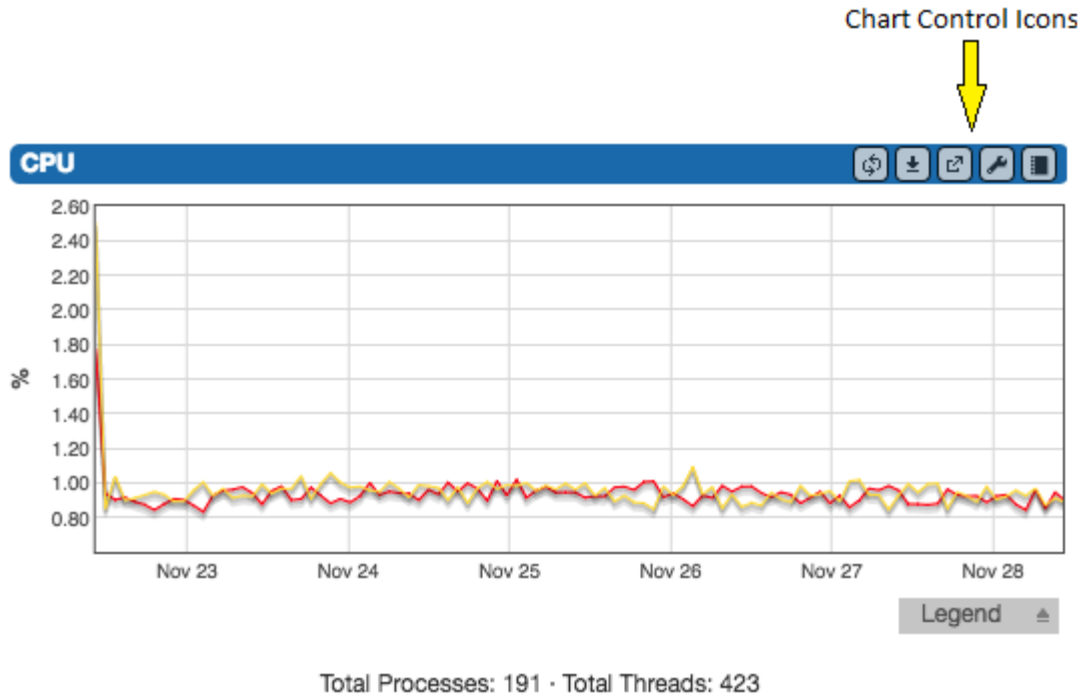

Figure 3.9 – The PEM Client chart control icons.

Select an icon to:

Refresh the content of a chart, graph or table.

Download an image of the chart or graph.

Expand the chart or graph to full-screen.

Personalize the chart, graph, or table settings for the current user.

View information about the chart, graph, or table.

For more information about customizing the graphics displayed on the PEM dashboards, please see the PEM client online help.

4 Managing a Monitored Server

Before you can manage or monitor a server with PEM, you must register the server with PEM, and bind an agent. A server may be bound to a remote agent (an agent that resides on a different host), but if the agent does not reside on the same host, it will not have access to all of the statistical information about the instance.

4.1 Manually Registering a Server

To manage or monitor a server with PEM, you must:

- register your Advanced Server or PostgreSQL server with the PEM server.

- bind the server to a PEM agent.

You can use the `Create - Server` dialog to provide registration information for a server, bind a PEM agent, and display the server in PEM client tree control. To open the `Create - Server` dialog, navigate through the `Create` option on the `Object` menu (or the context menu of a server group) and select `Server…` (see Figure 4.1).

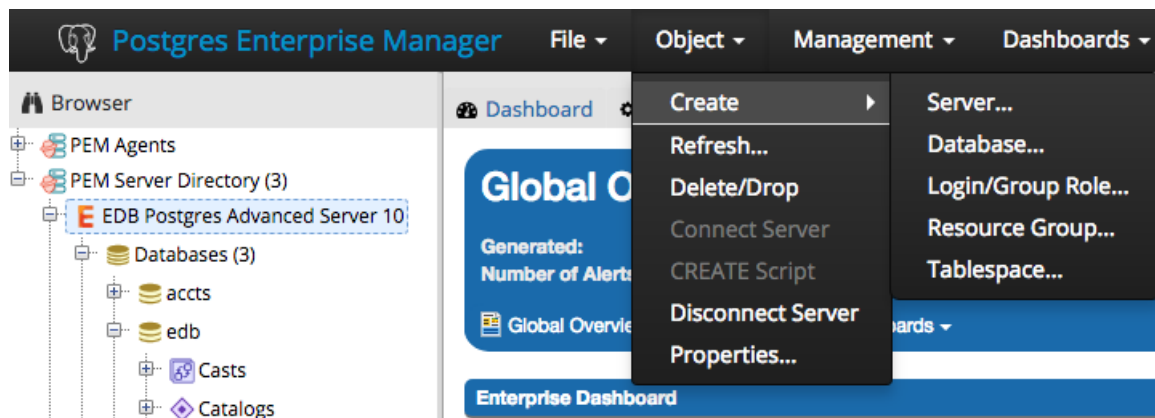
Figure 4.1 – Accessing the Create – Server dialog.

Please note: you must ensure the `pg_hba.conf` file of the Postgres server that you are registering allows connections from the host of the PEM client before attempting to connect.
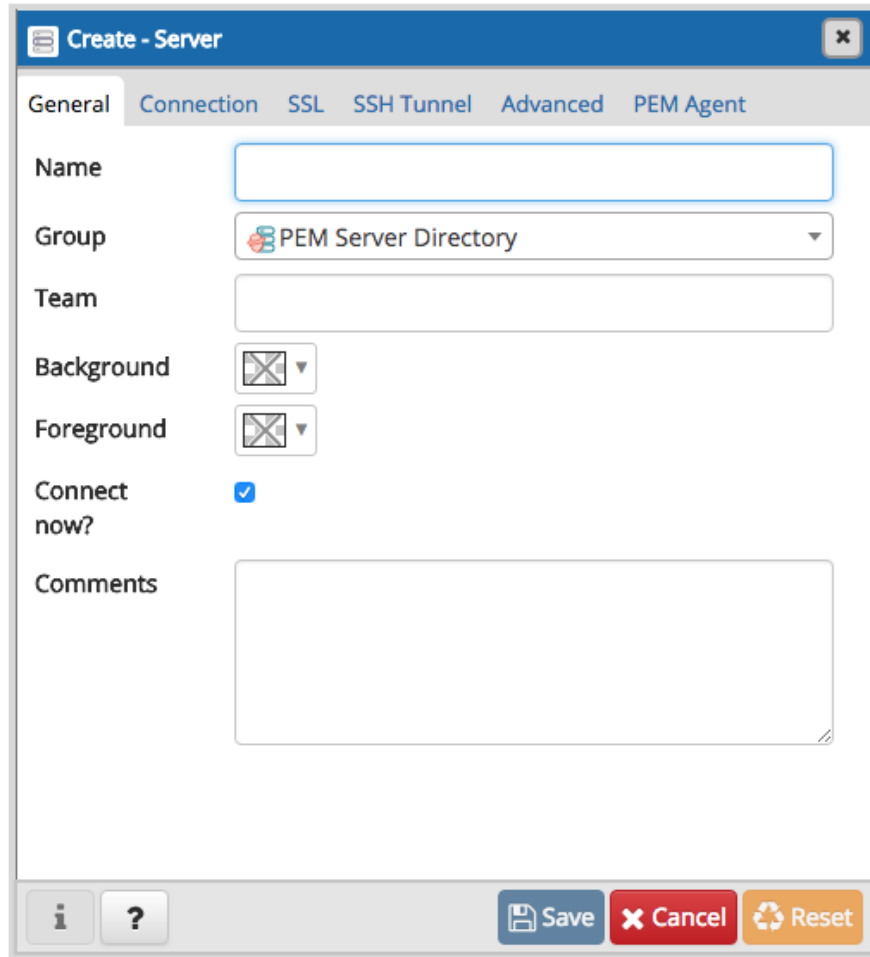
Figure 4.2 – The General tab of the Create – Server dialog.

Use the fields on the `General` tab (shown in Figure 4.2) 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 `Browser` tree control.

- You can use groups to organize your servers and agents in the 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 list box to select the server group in which the new server will be displayed.

- Use the `Team` field to specify a Postgres 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 will see the server.

- 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 server connection when you click the `Save` button. Leave `Connect now?` unchecked if you do not want the PEM client to validate the specified connection parameters until a later connection attempt.

- Provide notes about the server in the `Comments` field.

Figure 4.3 – The Connection tab of the Create – Server dialog.

Use fields on the `Connection` tab (shown in Figure 4.3) to specify connection details for the server:

24

- 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 be expected to contain `pgAgent` schema and `adminpack` objects installed (both optional). On PostgreSQL 8.1 and above, the maintenance DB is normally called `postgres`; on earlier versions `template1` is often used, though it is preferrable to create a `postgres` database to avoid cluttering the template database.

- Specify the name that will be used when authenticating with the server in the `Username` field.

- Provide the password associated with the specified user in the `Password` field.

- Check the box next to `Save password?` to instruct PEM to store passwords in the `~/.pgpass` file (on Linux) or `%APPDATA%\\postgresql\\pgpass.conf` (on Windows) for later reuse. For details, see the `pgpass` documentation. Stored passwords will be used for all libpq based tools. To remove a password, disconnect from the server, open the server's Properties dialog and uncheck the selection.

- 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.
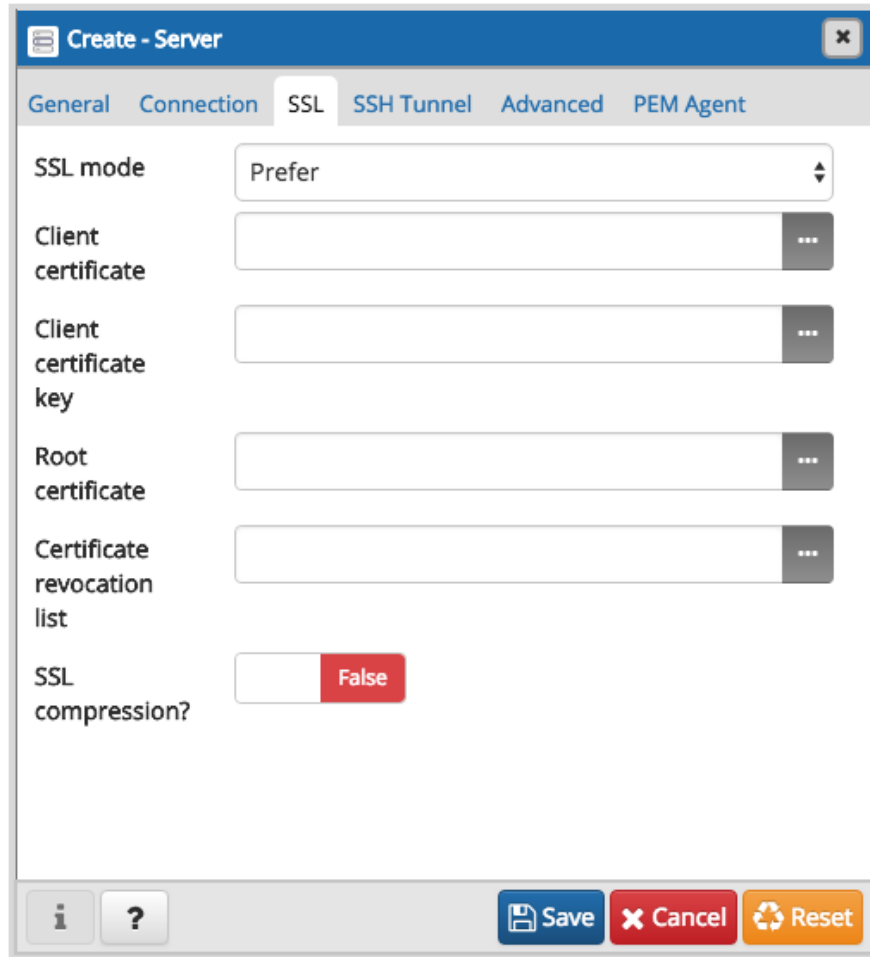
Figure 4.4 – The SSL tab of the Create – Server dialog.

Use the fields on the `SSL` tab (see Figure 4.4) 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 the PostgreSQL documentation at:

  https://www.postgresql.org/docs/current/static/libpq-ssl.html

You can use the platform-specific `File` manager dialog to upload files that support SSL encryption to the server. To access the File manager, 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 `~/.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 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 `~/.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 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` file. 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: 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.
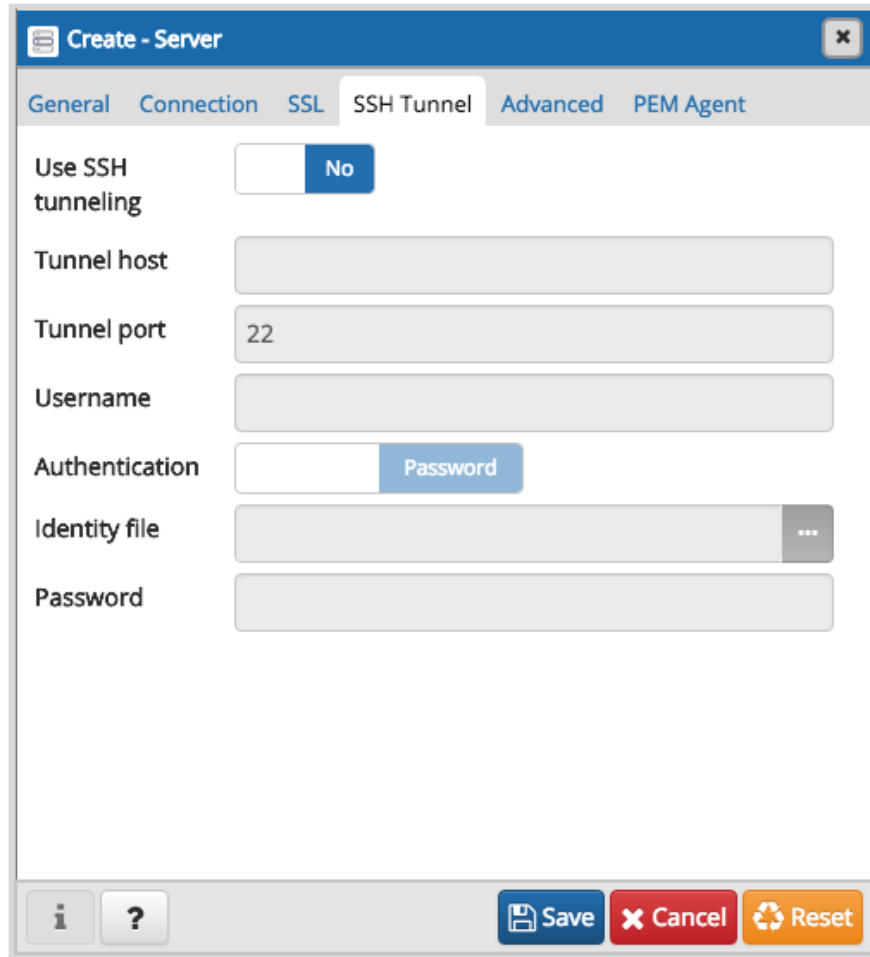
Figure 4.5 – The SSH Tunnel tab of the Create – Server dialog.

Use the fields on the `SSH Tunnel` tab (see Figure 4.5) 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.

Figure 4.6 – The Advanced tab of the Create – Server dialog.

Use fields on the `Advanced` tab (shown in Figure 4.6) 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. Note that you can also limit the schemas shown in the database from the database properties dialog 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 login without providing a password when they connect. For more information, see the Postgres documentation at:

    http://www.postgresql.org/docs/current/static/libpq-pgpass.html

    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 Linux machines, this is the name of the `init` script used to start the server in `/etc/init.d`. For example, the name of the Advanced Server 10 service is `edb-as-10`. For local servers, the setting is operating system dependent:

    o If the PEM client is running on a Windows machine, it can control the postmaster service if you have sufficient 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.

    o 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 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 master 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.
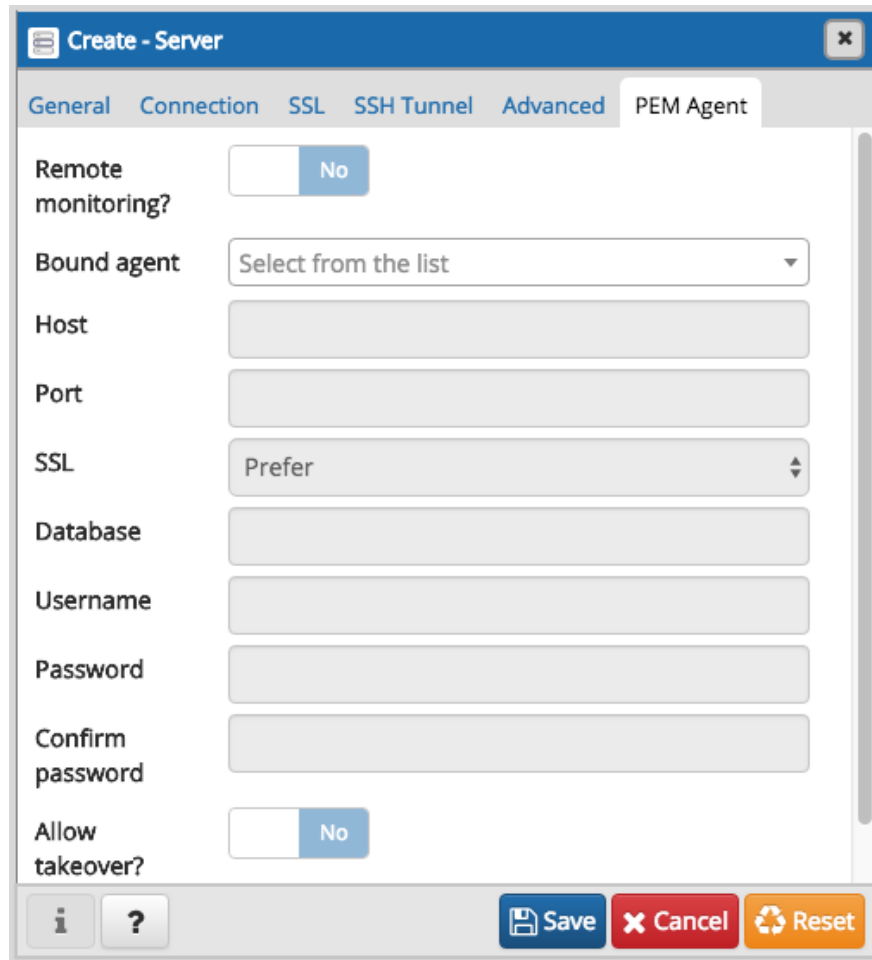
Figure 4.7 – The PEM Agent tab of the Create – Server dialog.

Use fields on the `PEM Agent` tab (shown in Figure 4.7) to specify connection details for the PEM agent:

- Move the `Remote monitoring?` slider to `Yes` 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. For more information about using SSL encryption, see the PostgreSQL documentation at:

  http://enterprisedb.com/docs/en/10/pg/libpq-ssl.html

- Use the `Database` field to specify the name of the database to which the agent will initially connect.

- Specify the name of the role that agent should use when connecting to the server in the `User name` 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; for example, you can use a `.pgpass` file.

- Set the `Allow takeover?` slider to `Yes` 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.

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.

4.2 Automatic Server Discovery

If the server you wish to monitor resides on the same host as the monitoring agent, you can use the `Auto Discovery` dialog to simplify the registration and binding process.

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 (see Figure 4.8), highlight the name of a PEM agent in the PEM client tree control, and select `Auto Discovery...` from the `Management` menu.

Figure 4.8 – The PEM Auto Discovery dialog.

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 connection properties for the agent the `Agent Connection Details` box.

Use the `Check All` button to select the box next to all of the displayed servers, or `Uncheck All` to deselect 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 `User name` 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 register the servers. Click `Cancel` to exit without preserving any changes.
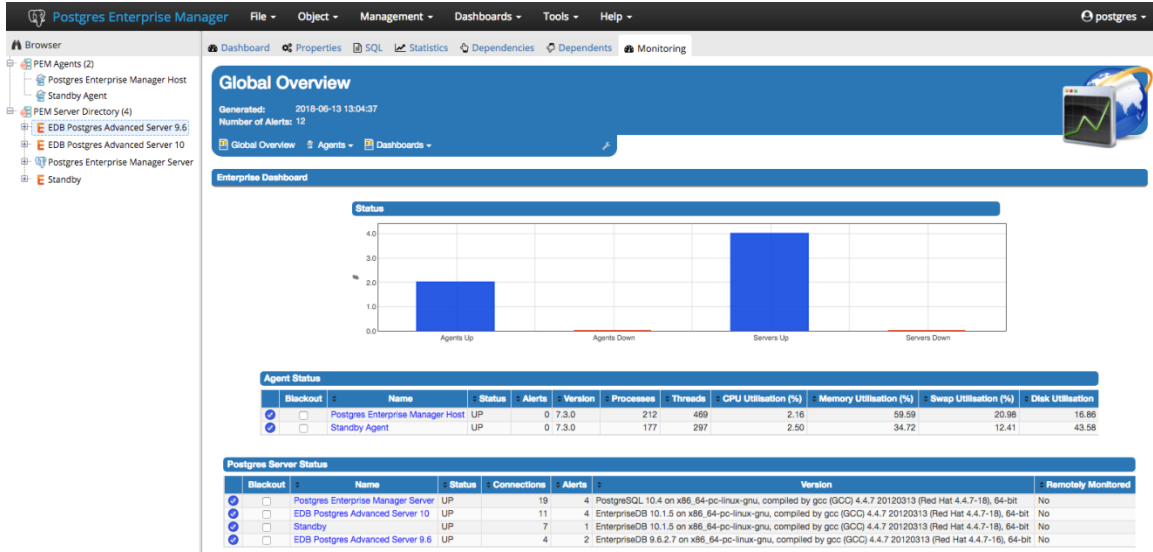
Figure 4.9 – The registered server.

After clicking the `OK` button, the newly registered server is displayed in the PEM tree control (see Figure 4.9) and is monitored by the PEM server.

4.3 Using the pemworker Utility to Register a Server

You can use the pemworker utility to register a server for monitoring by the PEM server or to unregister a database server. During registration, the pemworker utility will bind 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
```

on a Windows host, use the command:

```
pemworker.exe REGISTER-SERVICE
```

Append command line options to the command string when invoking the pemworker utility. Each option should be followed by a corresponding value:

Option	Description
`--pem-user`	Specifies the name of the PEM administrative user. Required.
`--server-addr`	Specifies the IP address of the server host, or the fully qualified domain name. 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 /. Required.
`--server-port`	Specifies the port number of the host. Required.
`--server-database`	Specifies the name of the database to which the server will connect. Required.
`--server-user`	Specify the name of the user that will be used by the agent when monitoring the server. Required.
`--server-service-name`	Specifies the name of the database service that controls operations on the server that is being registered (STOP, START, RESTART, etc.). Optional.
`--remote-monitoring`	Include the --remote-monitoring clause and a value of false (the default) to indicate that the server is installed on the same machine as the PEM agent. When remote monitoring is enabled (true), 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). Required.
`--efm-cluster-name`	Specifies the name of the Failover Manager cluster that monitors the server (if applicable). Optional.
`--efm-install-path`	Specifies the complete path to the installation directory of Failover Manager (if applicable). Optional.
`--asb-host-name`	Specifies the name of the host to which the agent is connecting.
`--asb-host-port`	Specifies the port number that the agent will use when connecting to the database.
`--asb-host-db`	Specifies the name of the database to which the agent will connect.
`--asb-host-user`	Specifies the database user name that the agent will supply when authenticating with the database.
`--asb-ssl-mode`	Specifies the type of SSL authentication that will be used for

	connections. Supported values include: prefer, require, disable, verify-CA, verify-full.
`--group`	Specifies the name of the group in which the server will be displayed.
`--team`	Specifies the name of the group role that will be allowed to access the server.
`--owner`	Specifies the name of the role that will own the monitored server.

When registering a database server, you must provide the password associated with the administrative user on the new server. Use the `PEM_NEW_SERVER_PASSWORD` environment variable to provide the associated password.

Before any changes are made on the PEM database, the connecting agent is authenticated with the PEM database server. When invoking the pemworker utility, you must provide the password associated with the PEM server administrative user. There are three ways to specify the administrative password; you can:

- set the `PEM_MONITORED_SERVER_PASSWORD` environment variable.
- provide the password on the command line with the `PGPASSWORD` keyword.
- create an entry in the `.pgpass` file.

Failure to provide the password will result in a password authentication error. The PEM server will acknowledge that the server has been registered properly.

4.3.1 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-SERVICE
```

Append command line options to the command string when invoking the pemworker utility. Each option should be followed by a corresponding value:

Option	Description
--pem-user	Specifies the name of the PEM administrative user. Required.
--server-addr	Specifies the IP address of the server host, or the fully qualified domain name. 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 /. Required.
--server-port	Specifies the port number of the host. Required.

When invoking the pemworker utility, you must provide the password associated with the PEM server administrative user. There are three ways to specify the password of the administrative user; you can:

- 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.

Failure to provide the password will result in a password authentication error. The PEM server will acknowledge that the server has been unregistered.

4.4 Verifying the Connection and Binding

Once registered, the new server will be added to the PEM `Browser` tree control, and be displayed on the `Global Overview` (see Figure 4.10).
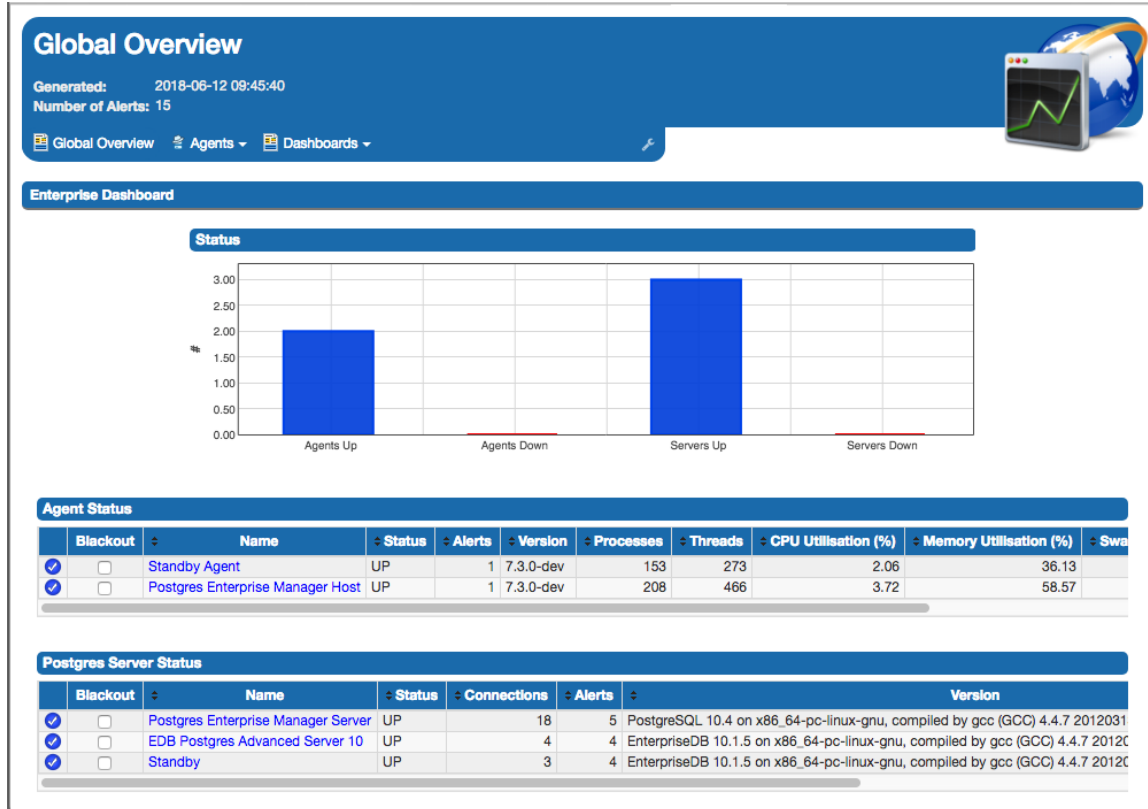

Figure 4.10 - The Global Overview dashboard.

When initially connecting to a newly bound server, the `Global Overview` dashboard may 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, which may take a few minutes to complete depending on network availability.

Within a few minutes, bar graphs on the `Global Overview` dashboard should show that the agent has now connected successfully, and the new server is included in the `Postgres Server Status` list.

If after five minutes, the `Global Overview` dashboard still does not list the new server, you should review the log files for the monitoring agent, checking for errors. Right-click the agent's name in the tree control, and select the `Probe Log Analysis` option from the `Dashboards` sub-menu of the context menu.

4.5 Remotely Starting and Stopping Monitored Servers

PEM allows you to startup and shutdown managed server instances with the PEM client. To configure a server to allow PEM to manage the service, complete the `Server registration` dialog, registering the database server with a PEM agent and:

- specify the `Store on PEM Server` option on the `Properties` dialog.

- specify the name of a service script in the `Service ID` field on the `Advanced` tab:

  - For Advanced Server 10, the service name is `edb-as-10`.
  - For Advanced Server 9.6, the service name is `edb-as-9.6`.
  - For Advanced Server 9.5 and earlier, the service name is `ppas-9.`*x*.
  - For PostgreSQL 10, the service name is `postgresql-10`.
  - For PostgreSQL 10, the service name is `postgresql-9.`*x*.

After connecting to the server, you can start or stop the server by highlighting the server name in the tree control, and selecting `Queue Server Startup` or `Queue Server Shutdown` (shown in Figure 4.11) from the `Management` menu.
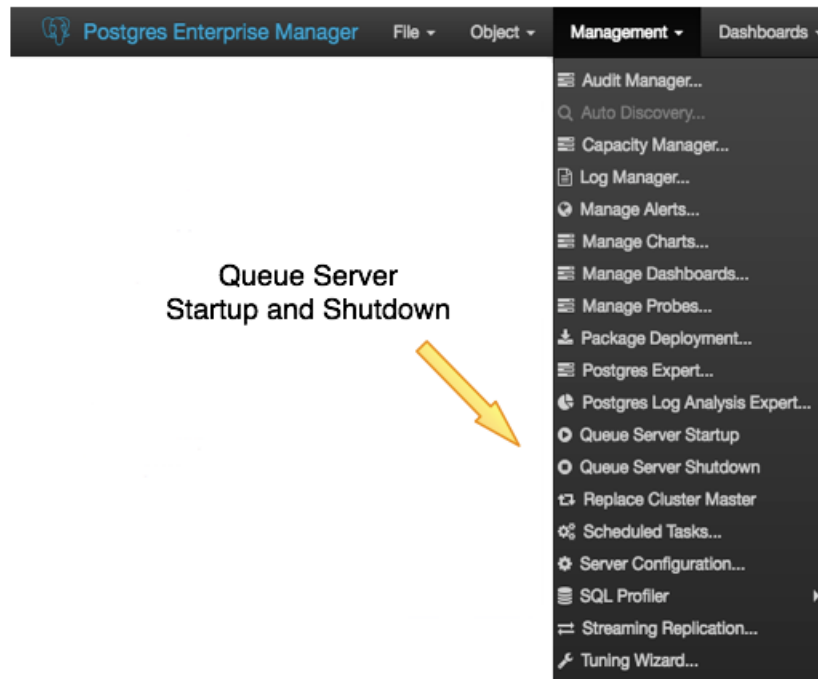

Figure 4.11 - The Management menu of a managed server.

5 General Database Administration

The chapter covers some of the basic administrative tasks that can be carried out with the PEM client.

5.1 Editing the PEM Server Configuration

You can use the PEM client to graphically manage the configuration parameters of the PEM server to enable features or modify default settings. To open the `Server Configuration` dialog (see Figure 5.1), select `Server Configuration...` from the `Management` menu.

Figure 5.1 - The Server Configuration dialog.

To modify a parameter value, edit the content displayed in the `Value` field to the right of a parameter name. Click the `Save` button to preserve your changes, or click the `Close` button to exit the dialog without applying the changes. Use the `Reset` button to return the parameters to their original value.

5.2 Starting and Stopping the PEM Server and Agents

The PEM server starts, stops and restarts when the Postgres server instance on which it resides starts, stops or restarts; use the same commands to control the PEM server that you would use to control the Postgres server.

The PEM agent is controlled by a service named `pemagent`.

On Linux platforms, the command that stops and starts the service script will vary by platform and OS version.

The Windows operating system includes a graphical service controller that displays the server status, and offers point-and-click server control. The `Services` utility 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 service name (see Figure 5.2).
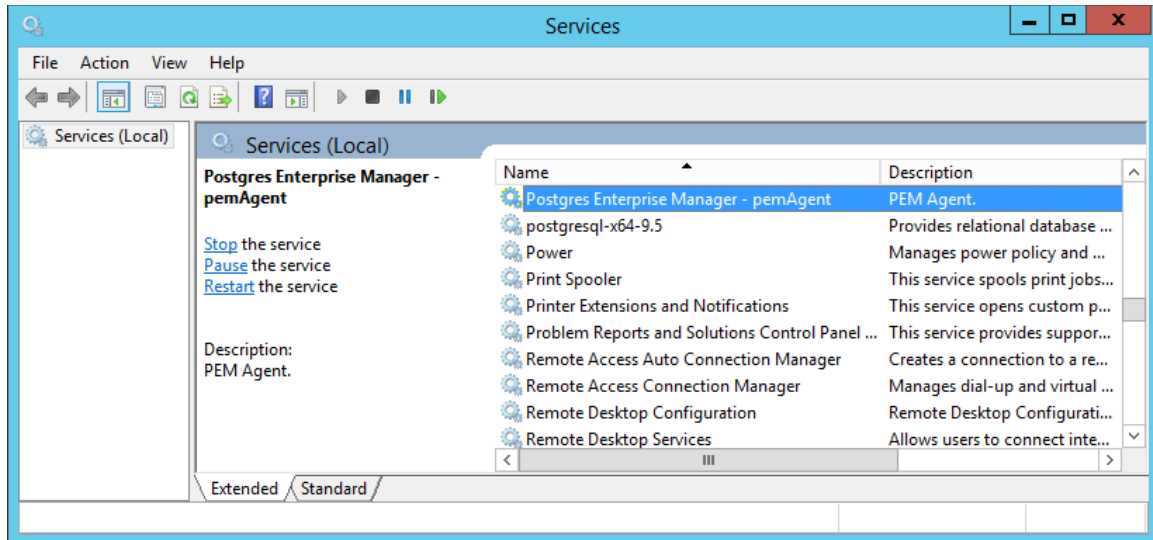
Figure 5.2 – The PEM service in the Windows Services window.

Use the `Stop`, `Pause`, `Start`, or `Restart` buttons to control the state of the service.

Please note that any user (or client application) connected to the Postgres server will be abruptly disconnected if you stop the service.

For more information about controlling a service, please consult the EDB Postgres Advanced Server Installation Guide, available from the EnterpriseDB website at:

https://www.enterprisedb.com/resources/product-documentation

5.3 Managing Security

PEM provides a graphical way to manage your Postgres roles and servers.

5.3.1 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 to create a role, expand the node for the server on which the role will reside in the PEM tree control, and right-click on the `Login/Group Roles` node to access the context menu. Then, select `Login/Group Role…` from the `Create` menu (see Figure 5.3).
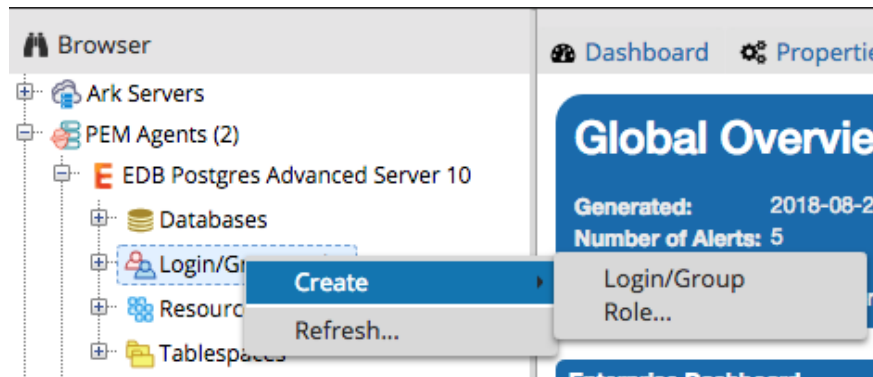

Figure 5.3 - The context menu of the Login Roles node

Use fields on the tabs of the `Create - Login/Group Role` dialog (see Figure 5.4) to define the role. To display the PEM online help in a browser tab, click the help (`?`) button located in the lower-left corner of the dialog.

When you've finished defining the new role, click `Save` to create the role.
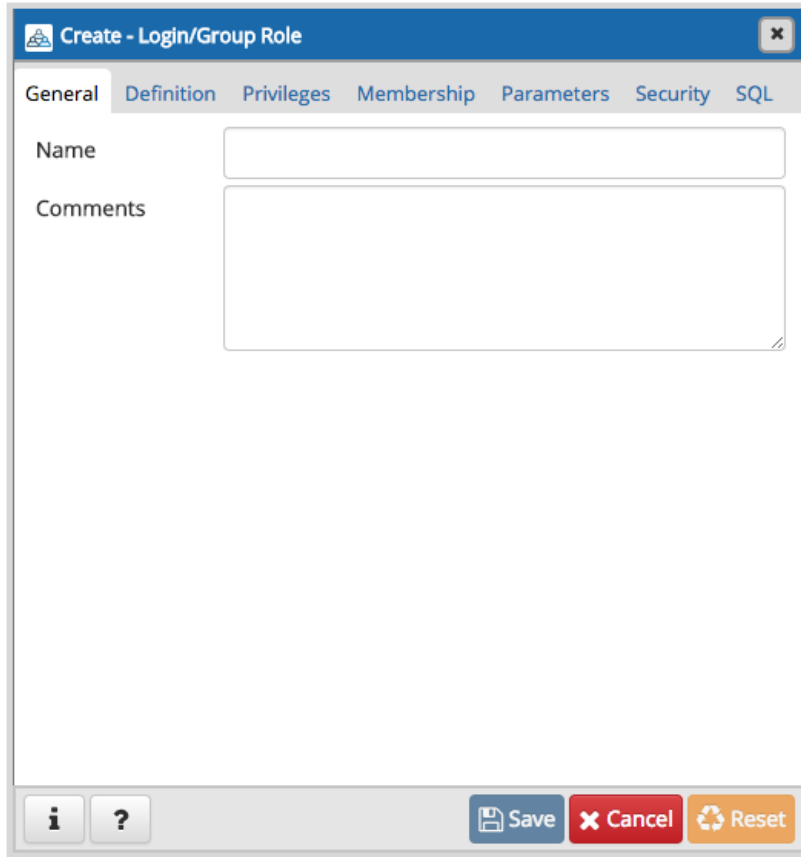
Figure 5.4 - The Login Role dialog

To modify the properties of an existing login role, right click on the name of a login role in the tree control, and select `Properties` from the context menu. To delete a login role, right click on 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:

http://www.postgresql.org/docs/10/static/sql-createrole.html

44

5.3.2 Group Roles

Group roles can serve as containers, used to dispense system privileges (such as creating databases) and object privileges (e.g. inserting data into a particular table). The primary purpose of a group role is to make the mass management of system and object permissions much 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 control) to create and manage group roles. Options on the context menu provide access to a dialog that allows you to create a new role or modify the properties of an existing role. You can find more information about creating roles at:

http://www.postgresql.org/docs/10/static/sql-createrole.html

5.3.3 Using PEM Pre-Defined Roles to Manage Access to PEM Functionality

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.
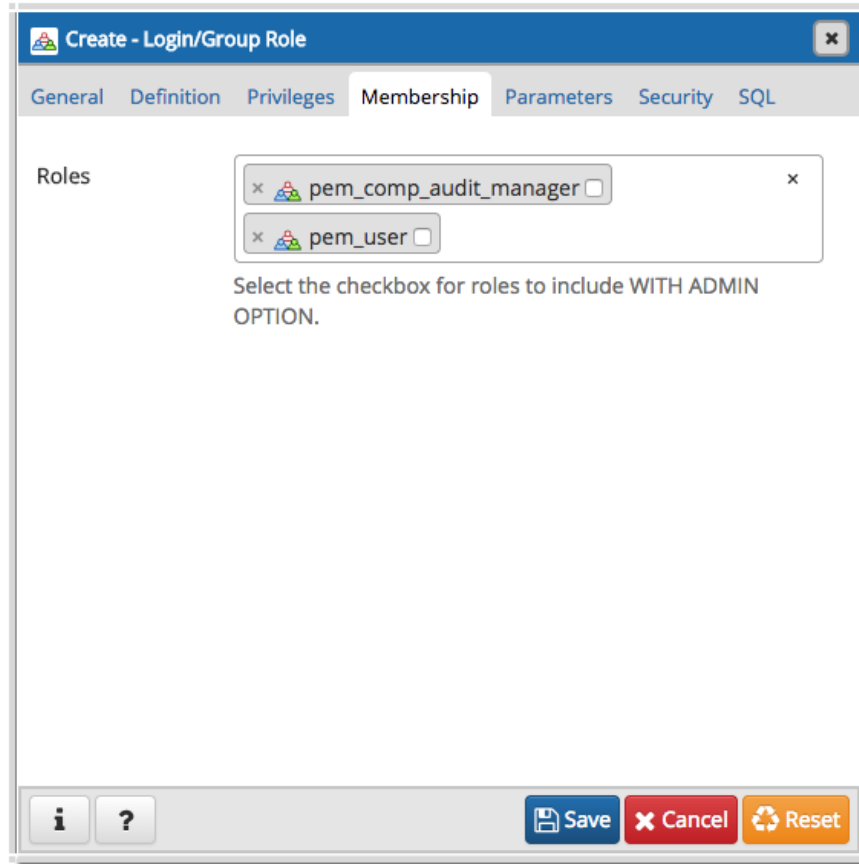
Figure 5.5 - The Membership tab.

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.

The `SQL` tab displays the SQL command that the server will execute when you click `Save`.

Figure 5.6 - The SQL tab.

The example shown above creates a login role named `acctg_clerk` 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
pem_super_admin		Role to manage/configure everything on Postgres Enterpise Manager.
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 Postgres Enterprise Manager.
pem_component	pem_admin	Role to run/execute all wizard/dialog 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).

47

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, pem_manage_alert	Role for configuring the alerts on any monitored objects.
pem_manage_probe	pem_admin	Role to create, update, delete the custom probes, and change custom probe configuration.
pem_config_probe	pem_config, 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 functionality.
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_package_deployment	pem_component	Role to run the Package Deployment Wizard.
pem_comp_streaming_replication	pem_component	Role to run the Streaming Replication Wizard.
pem_comp_tuning_wizard	pem_component	Role to run the Tuning Wizard.

5.3.4 Using a Team Role

When you register a server for monitoring by PEM, you can specify a *Team* that will be associated with the server. A Team is a group role that can be used to allow or restrict access to one or more monitored servers to a limited group of role members. The PEM client will only display a server with a specified `Team` 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, expand the node for the server on which the role will reside in the PEM tree control, and right-click on the `Login/Group Roles` node to access the context menu. Then, select `Login/Group Role…` from the `Create` menu; when the `Create - Login/Group Role` dialog opens, use the fields provided to specify the properties of the team role.

5.3.5 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 graphical interface of the PEM client.

Object permissions are managed via the graphical object editor for each particular object. For example, to assign privileges to access a database table, right click on the table name in the tree control, and select the `Properties` option from the context menu. Use the options displayed on 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.

5.4 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 contained within it. The PEM client provides dialogs that allow you to create and manage databases, and all of the various objects that comprise a database (e.g. tables, indexes, stored procedures, etc.).

Creating a database is easy in PEM: simply right click on any managed server's `Databases` node and select `Database...` from the `Create` menu. After defining a database, you can create objects within the new database.

For example, to create a new table, right click on a `Tables` node, select `Table...` from the `Create` menu. When the `New Table` dialog opens, specify the attributes of the new table (see Figure 5.7).
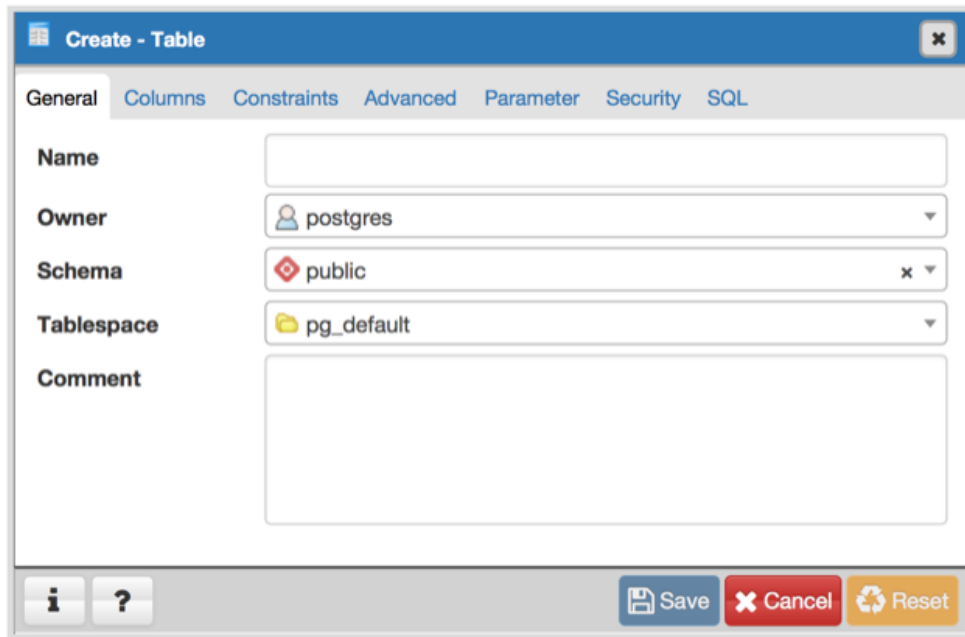

Figure 5.7 - Use PEM's dialogs to create and manage database objects.

PEM provides similar dialogs for the creation and management of other database objects:

- tables
- indexes
- stored procedures
- functions
- triggers
- views
- constraints, etc.

Each object type is displayed in the tree control; right click on the node that corresponds to an object type to access the `Create` menu and create a new object, or select `Properties` from the context menu of a named node to perform administrative tasks for the highlighted object.

5.5 Online Help and Documentation

PEM contains built-in help that provides assistance in using the tool (see Figure 5.8). To access the online help for PEM, select `Online Help` from the `Help` option on the `Help` menu.

Figure 5.8 – PEM online help.

The `Help` menu also allows quick access to the EnterpriseDB website.

6 Server Management

The sections that follow provide information about modifying the `pg_hba.conf` file, restarting the PEM server, and controlling the HTTPD service on Linux and Windows.

6.1 Controlling the PEM Server or PEM Agent on Linux

On Linux platforms, the name of the service script that controls:

- A PEM server on Advanced Server 10 is `edb-as-10`.
- A PEM server on Advanced Server 9.6 is `edb-as-9.6`.
- A PEM server on Advanced Server 9.5 or prior is `ppas-9.`*x*, where *x* is the version number.
- A PEM server on PostgreSQL 10 is `postgresql-10`
- A PEM server on PostgreSQL 9.*x* is `postgresql-9.`*x*.
- A PEM agent is `pemagent`.

You can use the service script to control the service.

If you are using RHEL or CentOS 6.x, open a command line, assume superuser privileges, and enter:

```
/etc/init.d/service_name action
```

To control a service on RHEL or CentOS version 7.x, open a command line, 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.

6.2 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 point-and-click 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-9.x`
- a PEM agent is `Postgres Enterprise Manager - pemAgent`

Where x indicates the server version number.

Navigate through the Windows `Control Panel` to open the `Services` utility. When the utility opens, use the scroll bar to browse the list of services (see Figure 6.1).
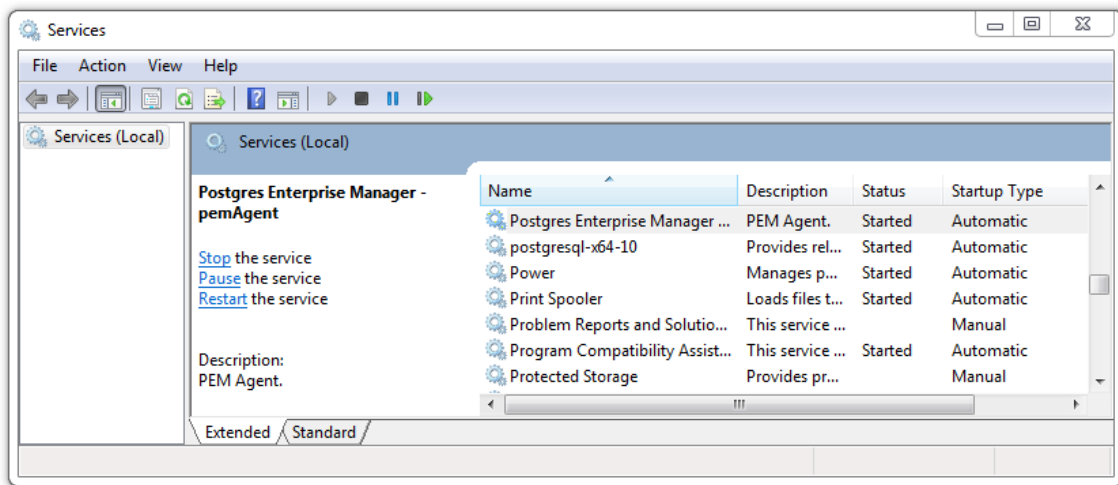
Figure 6.1 – The Windows Services window.

Use the `Stop the service` option to stop a service. Any user (or client application) connected to the server will be abruptly disconnected if you stop the service.

Use the `Pause the service` option to instruct Postgres to reload a service's configuration parameters. The `Pause the service` option is an effective way to reset parameters without disrupting user sessions for many of the configuration parameters.

Use the `Start the service` option to start a service.

6.3 Controlling the HTTPD Server

On Linux, you can confirm the status of the PEM-HTTPD service by opening a command line, and entering the following command:

```
ps -ef | grep httpd
```

If Linux responds with an answer that is similar to the following example, `httpd` is not running:

```
user      13321 13267   0 07:37 pts/1     00:00:00 grep httpd
```

Use the following command to start the service on a CentOS or RHEL 6.x system:

```
/etc/init.d/httpd start
```

On a CentOS or RHEL 7.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, scroll through the list to locate the `PEM HTTPD` service (see Figure 6.2).
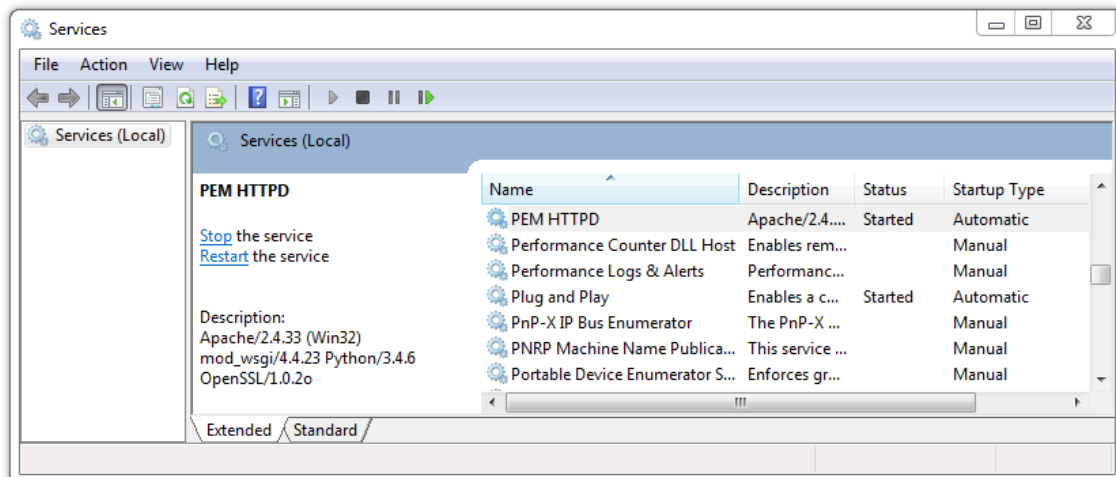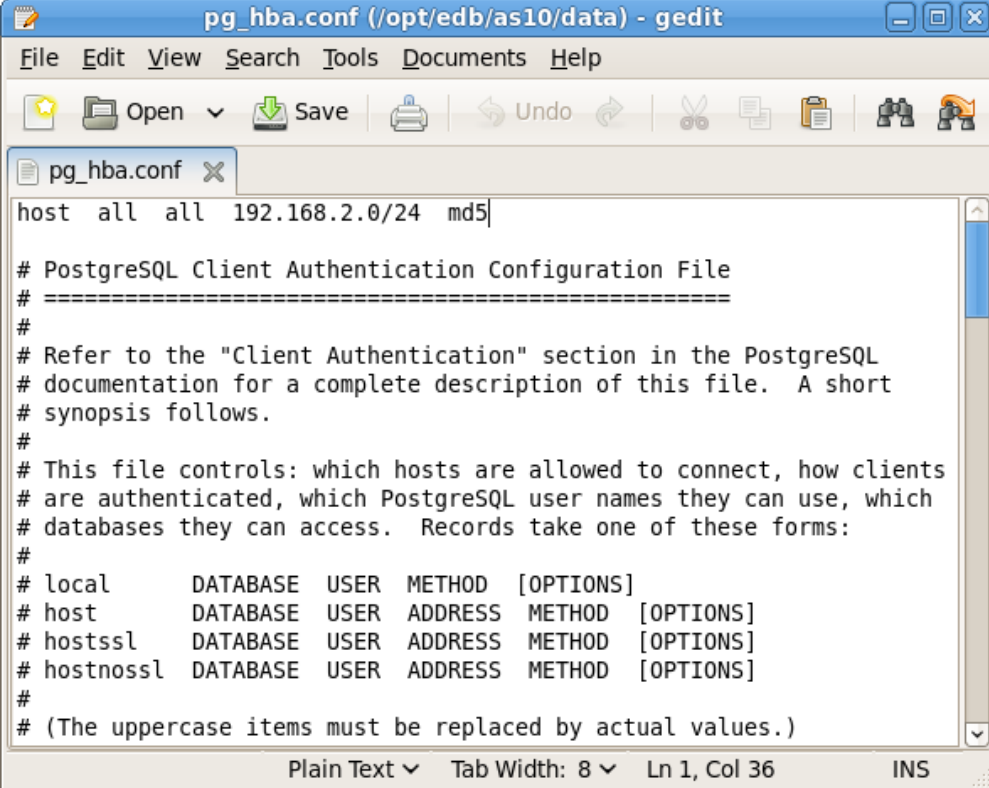
Figure 6.2 - The PEM HTTPD Windows service.

The `Status` column displays the current state of the server. Click the `Start` link to start PEM HTTPD if the service is not running.

6.4 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.

During the PEM server installation process, you are prompted for the IP address and connection information for hosts that will be monitored by PEM; this information is added to the top of the `pg_hba.conf` file of the PEM backing database (see Figure 6.3).

Figure 6.3 - PEM entries in the pg_hba.conf file.

You may 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 does not reside 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 Advanced Server 10 host, the default location of the `pg_hba.conf` is:

```
/opt/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 on which the `pg_hba.conf` file 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 may specify the address of a network host, or a network address range. For example, if you wish 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, please see the PostgreSQL core documentation at:

http://www.postgresql.org/docs/10/static/auth-pg-hba-conf.html

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 (see Figure 6.4).
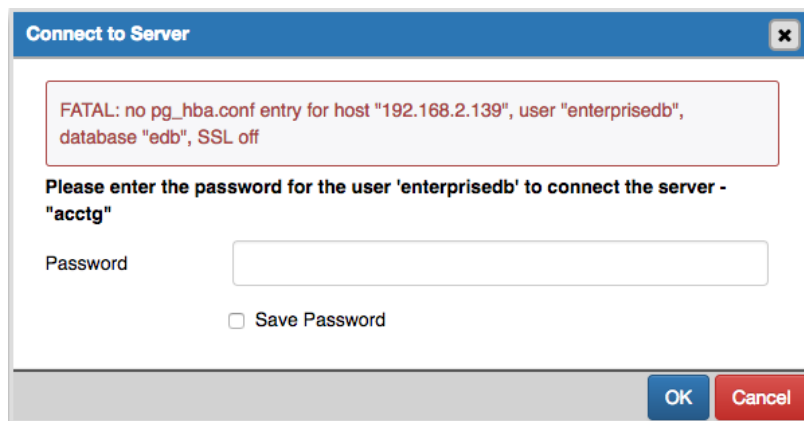
Figure 6.4 – Connect to Server error.

If you receive this error when connecting to the database server, modify the `pg_hba.conf` file, adding an entry that allows the connection.

6.5 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 within the configuration file specify who may connect to a specific database, and 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 would take the form:

```
host   all   postgres   127.0.0.1/32   md5
```

Depending on your system's configuration, you may also need to create a password file for the user account that the PEM agent uses to connect to the server, to allow 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` would take 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, visit the EnterpriseDB website at:

http://www.postgresql.org/docs/10/static/libpq-pgpass.html

For more information about the authentication methods supported by Postgres, see the PostgreSQL core documentation at:

http://www.postgresql.org/docs/10/static/client-authentication.html

6.6 Modifying PEM to Use a Proxy Server

If your network configuration prevents direct communication between PEM and the EnterpriseDB website, you can configure a proxy server for use by PEM when:

- updating the `package_catalog` table with information about the packages that are available for installation or update
- reading package options
- downloading packages

After configuring a proxy server on your network, modify the PEM server configuration, specifying the connection properties of the proxy, and instructing PEM to use the proxy server.

Figure 6.6 - The PEM Server Configuration dialog.

To access the `Server Configuration` dialog and modify the server configuration, connect to the PEM web interface, and select `Server Configuration…` from the `Management` menu. The PEM Server Configuration dialog opens as shown in Figure 6.6.

To modify a parameter value, locate the parameter, and modify the parameter value in the `Value` column. Use the following PEM Server configuration parameters to specify connection details that allow PEM to connect to the proxy server:

- Use the `proxy_server` parameter to specify the IP address of the proxy server.

- Specify a value of `t` in the `proxy_server_authentication` parameter to indicate that the proxy server will require PEM to authenticate when connecting; specify `f` if authentication is not required.

- Specify a value of `t` in the `proxy_server_enabled` parameter if PEM is required to use a proxy server when retrieving the package list, or `f` if a proxy server is not configured.

- Use the `proxy_server_password` parameter to provide the password associated with the user specified in `proxy_server_username`.

- Specify the port number of the proxy server in the `proxy_server_port` parameter.

- Specify the user name that should be used when authenticating with the proxy server in the `proxy_server_username` parameter.

When you've finished updating the parameters required to configure the proxy server, click the `Save` icon in the upper-right corner of the dialog before closing the dialog.

6.7 Certificate Management

Files stored in the data directory of the PEM server backing database contain information that helps the PEM server utilize secure connections:

- ca_certificate.crt
- ca_key.key
- server.crt
- server.key
- root.crl
- root.crt

The PEM agent that is installed with the PEM server monitors the expiration date of the ca_certificate.crt file. When the certificate is about to expire, PEM will:

- Make a backup of the existing certificate files.
- Create new certificate files, appending the new CA certificate file to the root.crt file on the PEM server.
- Create a job that renews the certificate file of any active agents.
- Restart the PEM server.

When you uninstall an agent, the certificate associated with that agent will be added to the certificate revocation list (maintained in the root.crl file) to ensure that the certificate cannot be used to connect to the PEM server.

For detailed information about manually replacing certificate files, please see Section 8.

7 PEM Agent Management

The sections that follow provide information about the behavior and management of a PEM agent.

7.1 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`. 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.

Please note that if you limit the operating system privileges of the PEM agent, some of the PEM probes will not return information, and the following functionality may be affected:

Probe or Action	Operating System	PEM Functionality Affected
Data And Logfile Analysis	Linux/Windows	The Postgres Expert will be unable to access complete information.
Session Information	Linux	The per-process statistics will be incomplete.
PG HBA	Linux/Windows	The Postgres Expert will be unable to access complete information.
Service restart functionality	Linux/Windows	The Audit Log Manager, Server Log Manager, Log Analysis Expert and PEM may be unable to apply requested modifications.
Package Deployment	Linux/Windows	PEM will be unable to run downloaded installation modules.
Batch Task	Windows	PEM will be unable to run scheduled batch jobs in Windows.

If you restrict the database privileges of the PEM agent, the following PEM functionality may be affected:

Probe	Operating System	PEM Functionality Affected
Audit Log Collection	Linux/Windows	PEM will receive empty data from the PEM database.
Server Log Collection	Linux/Windows	PEM will be unable to collect

		server log information.
Database Statistics	Linux/Windows	The Database/Server Analysis dashboards will contain incomplete information.
Session Waits/System Waits	Linux/Windows	The Session/System Waits dashboards will contain incomplete information.
Locks Information	Linux/Windows	The Database/Server Analysis dashboards will contain incomplete information.
Streaming Replication	Linux/Windows	The Streaming Replication dashboard will not display information.
Slony Replication	Linux/Windows	Slony-related charts on the Database Analysis dashboard will not display information.
Tablespace Size	Linux/Windows	The Server Analysis dashboard will not display complete information.
xDB Replication	Linux/Windows	PEM will be unable to send xDB alerts and traps.

If the probe is querying the operating system with insufficient privileges, the probe may return a `permission denied` error.

If the probe is querying the database with insufficient privileges, the probe may 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 will be written to the log file that contains the name of the probe, the reason the probe failed, and a hint that will help 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`.

7.2 Agent Configuration

A number of user-configurable parameters and registry entries control the behavior of the PEM agent. You may be required to modify the PEM agent's parameter settings to enable some PEM functionality, such as the Streaming Replication wizard. After modifying values in the PEM agent configuration file, you must restart the PEM agent to apply any changes.

With the exception of the `PEM_MAXCONN` parameter, we strongly recommend against modifying any of the configuration parameters or registry entries listed below without first consulting EnterpriseDB support experts *unless* the modifications are required to enable PEM functionality.

- On 64 bit Windows systems, PEM registry entries are located in
  `HKEY_LOCAL_MACHINE\\Software\\Wow6432Node\\EnterpriseDB\\PE M\\agent`

- On Linux systems, PEM configuration options are stored in the `agent.cfg` file, located in `/opt/edb/pem/agent/etc`

Parameter Name	Description	Value
`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's identifier is '2', and so on.
`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\\Ro aming\\pem/agent.key By default on Linux, /root/.pem/agent.key
`AgentCrtPath (on Windows)` or `agent_ssl_crt (on`	The complete path to the PEM agent's certificate file.	By default, on Windows, C:\\Users\\user_name\\AppData\\Ro aming\\pem/agent.crt By default on Linux,

Linux)		/root/.pem/agent.crt
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 warning.
log_location (on Linux)	Specifies the location of the PEM worker log file.	/var/log/pem/worker.log
agent_log_location (on Linux)	Specifies the location of the PEM agent log file.	/var/log/pem/agent.log
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.
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.
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	When set to true, the SMTP email feature is enabled.	By default, set to true for the agent that resides on the host of the PEM server;

`enable_smtp(on Linux)`		false for all other agents.
`EnableSNMP (on Windows)` or `enable_snmp (on Linux)`	When set to true, the SNMP trap feature is enabled.	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.
`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.
`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.
`AllowPackageManagement (on Windows)` or `allow_package_management (on Linux)`	If set to TRUE, the Update Monitor and Package Management features are enabled.	By default, set to FALSE.
`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	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).

	and a new connection (to a different database) is required to satisfy a waiting request.	
`AllowBatchProbes (on Windows)` or `allow_batch_probes (on Linux)`	If set to TRUE, the user will be able to create batch probes using the custom probes feature.	By default, set to FALSE.
`HeartbeatConnection (on Windows)` or `heartbeat_connection (on Linux)`	When set to TRUE, a dedicated connection is used for sending the heartbeats.	By default, set to FALSE.
`AllowStreamingReplicat ion (on Windows)` or `allow_streaming_replic ation (on Linux)`	If set to TRUE, the user will be able to configure and setup streaming replication.	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\Local\Temp On Linux, set to /tmp.
`ConnectionCustomSetup (on Windows)` or `connection_custom_setu p (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

7.3 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.
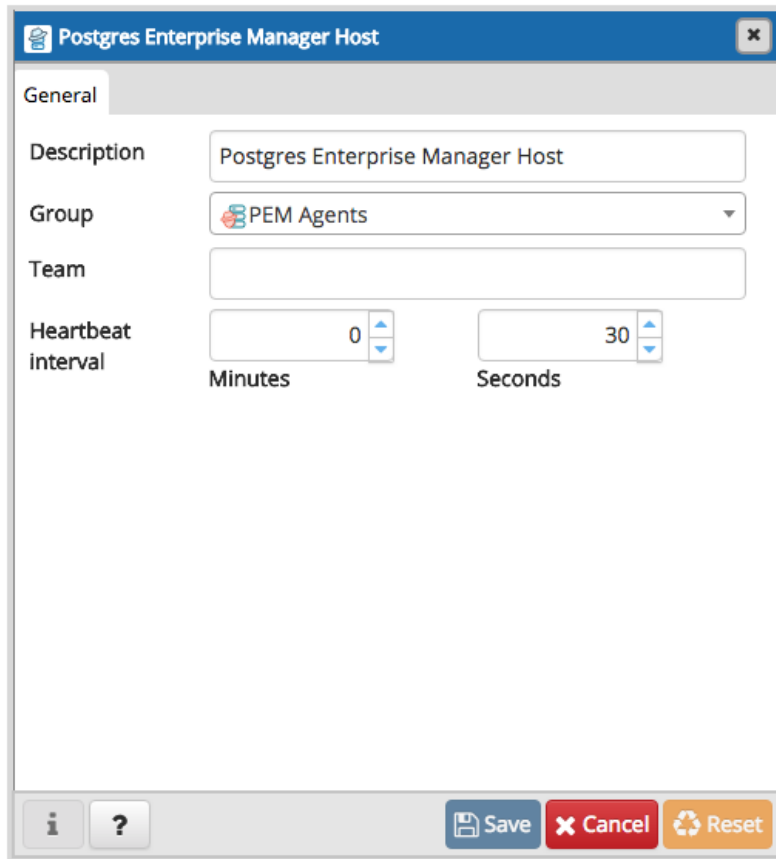
Figure 6.6 - The PEM Agent Properties dialog.

Use fields on the PEM Agent properties dialog 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 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 display 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.

8 Replacing SSL Certificates

The following steps detail replacing 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:

   ```
   /etc/init.d/pemagent stop
   ```

   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`; highlight the service name in the `Services` dialog, and click `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:

   ```
   /opt/PostgreSQL/10/data
   ```

   Make a copy of 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, the command:

   ```
   # cp ca_certificate.crt ca_certificate_old.crt
   ```

   creates a backup of the `ca_certificate` file with the word `old` appended to the entry.

3. Use the `openssl_rsa_generate_key()` function to generate the `ca_key.key` file.

```
/opt/PostgreSQL/10/bin/psql -U postgres -d pem --no-psqlrc
-t -A -c "SELECT public.openssl_rsa_generate_key(1024)" >
/opt/PostgreSQL/10/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 and modify the privileges on the `ca_key.key` file.

```
CA_KEY=$(cat /opt/PostgreSQL/10/data/ca_key.key)
```

```
chmod 600 /opt/PostgreSQL/10/data/ca_key.key
```

4. Use the key to generate the `ca_certificate.crt` file. For simplicity, place the SQL query into 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,
'/opt/PostgreSQL/10/data/ca_key.key')" > /tmp/_random.$$
```

Then use the variable to execute the query, placing the content into the `ca_certificate.crt` file.

```
/opt/PostgreSQL/10/bin/psql -U postgres -d pem --no-psqlrc
-t -A -f /tmp/_random.$$ >
/opt/PostgreSQL/10/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 /opt/PostgreSQL/10/data/ca_certificate.crt
```

```
rm -f /tmp/_random.$$
```

5. Re-use the `ca_certificate.crt` file as the `root.crt` file:

```
cp /opt/PostgreSQL/10/data/ca_certificate.crt
/opt/PostgreSQL/10/data/root.crt
```

Modify the permissions of the `root.crt` file:

```
chmod 600 /opt/PostgreSQL/10/data/root.crt
```

6. Use the `openssl_rsa_generate_crl()` function to create the certificate revocation list (`root.crl`).

```
/opt/PostgreSQL/10/bin/psql -U postgres -d pem --no-psqlrc
-t -A -c "SELECT
openssl_rsa_generate_crl('/opt/PostgreSQL/9.5/data/ca_certi
ficate.crt', '/opt/PostgreSQL/10/data/ca_key.key')" >
/opt/PostgreSQL/10/data/root.crl
```

Modify the permissions of the `root.crl` file:

```
chmod 600 /opt/PostgreSQL/10/data/root.crl
```

7. Use the `openssl_rsa_generate_key()` function to generate the `server.key` file.

```
/opt/PostgreSQL/10/bin/psql -U postgres -d pem --no-psqlrc
-t -A -c "SELECT public.openssl_rsa_generate_key(1024)" >>
/opt/PostgreSQL/10/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 /opt/PostgreSQL/10/data/server.key)
```

```
chmod 600 /opt/PostgreSQL/10/data/server.key
```

8. Use the `SSL_KEY` to generate the server certificate. Save the certificate in the `server.crt` file. For simplicity, 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'),
'/opt/PostgreSQL/10/data/ca_certificate.crt',
'/opt/PostgreSQL/10/data/ca_key.key')" > /tmp/_random.$$
```

Then, generate the `server.crt` file:

```
/opt/PostgreSQL/10/bin/psql -U postgres -d pem --no-psqlrc
-t -A -f /tmp/_random.$$ >>
/opt/PostgreSQL/10/data/server.crt
```

Modify the privileges on the `server.crt` file, and delete the temporary file:

```
chmod 600 /opt/PostgreSQL/10/data/server.crt
```

```
rm -f /tmp/_random.$$
```

9. Restart the Postgres server

```
/etc/init.d/postgresql-10 restart
```

8.1 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 is 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. Please note that 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 will 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:

```
/opt/PostgreSQL/10/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 `PEM/agent` section of the `/etc/postgres-reg.ini` file.

On Windows, the information is stored in the registry. On a 64-bit Windows installations, check:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\EnterpriseDB\PEM\agent
```

On a 32-bit Windows installation, check:

```
HKEY_LOCAL_MACHINE\SOFTWARE\EnterpriseDB\PEM\agent
```

2. After identifying the agents that will need key files, generate an `agent.key` for each agent. To generate the key, execute the following command, capturing the output in a file:

```
/opt/PostgreSQL/10/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:

```
/opt/PostgreSQL/10/bin/psql -U postgres -d pem --no-psqlrc
-t -A -c "SELECT
openssl_csr_to_crt(openssl_rsa_key_to_csr('$(cat
agent.key)', 'agent$ID', 'US', 'MA', 'Bedford', 'Postgres
Enterprise Manager', 'support@enterprisedb.com'),
'/opt/PostgreSQL/10/data/ca_certificate.crt',
'/opt/PostgreSQL/10/data/ca_key.key')" > agent.crt
```

Where $ID is the agent number of the agent (retrieved via the psql command line).

Modify the privileges of the `agent.crt` file:

```
chmod 600 agent.crt
```

4. Replace each agent's key and certificate file with the newly generated files before restarting the PEM agent service.

On Linux, restart the service with the command:

```
/etc/init.d/pemagent start
```

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`; highlight the service name in the `Services` dialog, and click `Start the service`.