



EDB Postgres Enterprise Manager[®]
Installation Guide
Version 7.8

June 05, 2019

EDB Postgres Enterprise Manager Installation Guide
by EnterpriseDB® Corporation
Copyright © 2013 - 2019 EnterpriseDB Corporation. All rights reserved.

EnterpriseDB Corporation, 34 Crosby Drive, Suite 201, Bedford, MA 01730, USA
T +1 781 357 3390 **F** +1 978 467 1307 **E** info@enterprisedb.com **www**.enterprisedb.com

EnterpriseDB, Postgres Plus, Postgres Enterprise Manager, and DynaTune are trademarks of EnterpriseDB Corporation.
Other names may be trademarks of their respective owners. © 2019.

Table of Contents

1	Introduction.....	5
1.1	What's New	6
1.2	Typographical Conventions Used in this Guide	7
2	Postgres Enterprise Manager™ - Overview	8
2.1	Supported Platforms.....	10
2.2	Hardware Pre-Requisites	11
2.3	Software Pre-Requisites.....	12
3	Installing Postgres Enterprise Manager™	14
3.1	Using a Graphical Installer to Install the PEM Server.....	15
3.1.1	Installing the PEM Server and PEM-HTTPD on the Same Host	17
3.1.2	Installing the PEM Server and PEM-HTTPD on Separate Hosts.....	27
3.1.2.1	Specifying a Database Host	33
3.1.2.2	Installing Web Services	41
3.1.3	Installing the PEM Server on an Existing Postgres Server.....	45
3.1.3.1	Preparing the Postgres Server	45
3.1.3.1.1	Installing the sslutils Extension.....	45
3.1.3.1.2	Creating a Service Script or Registering the Service.....	47
3.1.3.2	Invoking the PEM Server Installer	50
3.2	Using an RPM Package to Install the PEM Server.....	61
3.2.1	Installing on a SLES Host.....	66
3.2.2	Installing on a Debian or Ubuntu Host	67
3.3	Installing a PEM Agent.....	68
3.3.1	Using the Graphical Installer to Install an Agent.....	68
3.3.2	Installing an Agent with an RPM Package	77
3.3.3	Installing on a SLES Host.....	79
3.3.4	Registering an Agent.....	80
3.4	Invoking the Server or Agent Installers from the Command Line	82
3.4.1	Invoking the PEM Server Installer in Unattended Mode.....	83
3.4.2	Invoking the PEM Agent Installer in Unattended Mode	86
3.4.3	Invoking the PEM Server Installer in Text Mode.....	88
3.4.4	Invoking the PEM Agent Installer in Text Mode	92
4	The PEM Web Interface	95

5	Installing the SQL Profiler Plugin	98
5.1	Installing SQL Profiler.....	99
5.2	Configuring SQL Profiler	105
6	Upgrading and Uninstalling PEM Components	107
7	Reference - Linux Service Script.....	110

1 Introduction

Postgres Enterprise Manager (PEM) is designed to assist database administrators, system architects, and performance analysts when administering, monitoring, and tuning PostgreSQL and Advanced Server database servers. PEM has been designed to manage and monitor a single server or multiple servers from a single console, allowing complete control over monitored databases.

This document provides step-by-step instructions to guide you through the installation of Postgres Enterprise Manager™.

Throughout this guide, the term *Postgres* refers to either a PostgreSQL or an Advanced Server installation, where either is appropriate.

Language pack installers contain supported languages that may be used with EDB Postgres Advanced Server and EnterpriseDB PostgreSQL database installers. The language pack installer allows you to install Perl, TCL/TK, and Python without installing supporting software from third party vendors. For more information about installing and using Language Pack, please see the *EDB Postgres Language Pack Guide*, available from the EnterpriseDB Website.

Please note that in future PEM releases, a graphical installer will not be available for the PEM agent or server on a Linux host; you will be required to use an RPM package to install or update a PEM installation. For detailed information about using an RPM package to update the Linux host of a PEM server or agent that was installed with the graphical installer, please see the *PEM Upgrade and Migration Guide*.

1.1 What's New

The following features have been added to create Postgres Enterprise Manager 7.8:

- PEM 7.8 allows you to set the `enable_heartbeat_connection` to `true` during PEM agent registration; this creates a dedicated heartbeat connection between the PEM Agent and Server to constantly update the PEM server.
- The PEM 7.8 interface has improved accessibility features such as keyboard navigation enabled across dialogs and wizards, and new shortcut keys.
- PEM 7.8 adds capabilities that make the Query tool history persistent across sessions.
- Support has been added for the `IDENTITY` columns supported in PostgreSQL and EDB Postgres Advanced Server 10+.

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 preceding 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) consists of components that provide the management and analytical features of PEM:

- **PEM Server:** The PEM server is used as the data repository for monitoring data and as a server to which both agents and clients connect. The PEM server consists of an instance of PostgreSQL and an associated database for storage of monitoring data, and a server that provides web services.
- **PEM web interface:** The PEM web interface allows you to manage and monitor Postgres servers and utilize PEM extended functionality. The web interface software is installed with the PEM server installer, and is accessed via your choice of web browser.
- **PEM Agent:** The PEM agent is responsible for executing tasks and reporting statistics from the agent host and monitored Postgres instances to the PEM server. A single PEM agent can monitor multiple installed instances of Postgres that reside on one or many hosts.

The PEM Agent installer creates two executables: the PEM worker (`pemworker.exe`) and the PEM agent (`pemagent.exe`). Each PEM worker has a corresponding PEM agent that you can use to start or stop the PEM worker. The PEM agent will also restart the PEM worker should it terminate unexpectedly.

The PEM worker log file contains information related to PEM worker activity (probe activities, heartbeat responses, etc.), and is stored in `/var/log/pem/worker.log`.

- **SQL Profiler plugin:** This plugin to the Postgres server is used to generate the monitoring data used by the SQL Profiler tool. Installation of the SQL Profiler plugin is optional, but the plugin must be installed into each instance of Postgres you wish to profile. The SQL Profiler may be used with any supported version of an EnterpriseDB distribution of a PostgreSQL server or an Advanced Server (not just those managed through the PEM server). See Chapter 5 for details and supported versions.

The architectural diagram below (Figure 2.1) illustrates the relationship between the various servers and workstations involved in a typical PEM installation.

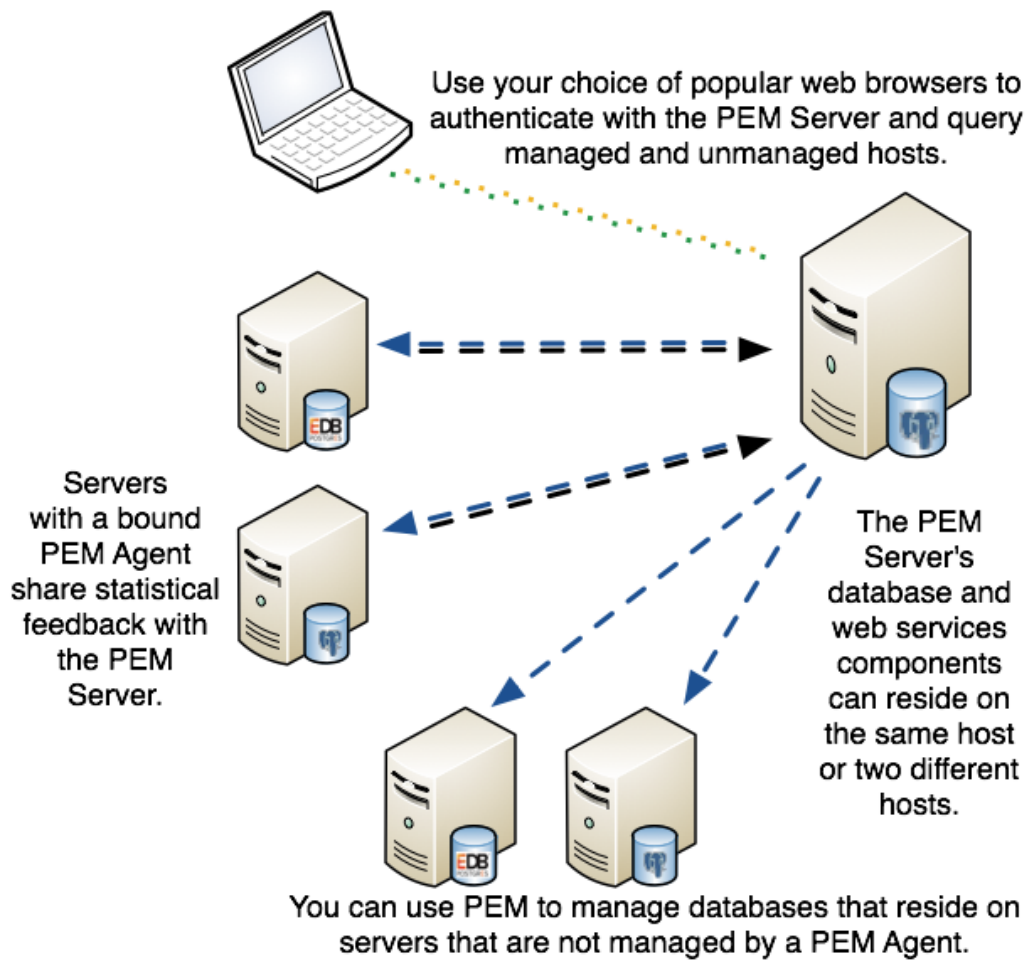


Figure 2.1 - A typical PEM installation.

2.1 Supported Platforms

The PEM server is supported on:

64-bit Windows:

- Windows Server 2012 R2
- Windows Server 2016
- Windows Server 2019

64-bit Linux:

- CentOS 6.x and 7.x
- OEL 6.x and 7.x
- Red Hat Enterprise Linux 6.x and 7.x
- SLES 12.x
- Debian 9.x (stretch) with .deb packages
- Ubuntu 18.04+ (bionic) with .deb packages

The PEM server is supported on these platforms, using Advanced Server or PostgreSQL backing databases (version 9.4 and higher).

The PEM agent is supported on any Linux or Windows platform on which Advanced Server or PostgreSQL version 9.4 or higher is supported.

The PEM web interface is supported on the following browsers:

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

PostgreSQL Version Support

PEM can manage and monitor:

- PostgreSQL versions 9.4 and higher
- Advanced Server 9.4 and higher

SQL Profiler

SQL Profiler for Postgres Enterprise Manager™ is supported on Advanced Server EnterpriseDB distributions of PostgreSQL version 9.4 and above.

2.2 Hardware Pre-Requisites

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

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

Additional disk space is required for data storage. Please note that resource usage will vary based on which probes are defined and enabled, and the activity level on the monitored databases. Monitoring server resources (as you use PEM) will let you know when you need to expand your initial system configuration.

2.3 Software Pre-Requisites

Modifying the `pg_hba.conf` File

The `pg_hba.conf` file manages connections for the Postgres server. You must ensure that the `pg_hba.conf` file on each monitored server allows connections from the PEM server, the monitoring PEM agent, and the host of the PEM-HTTPD server.

For information about modifying the `pg_hba.conf` file, see the *PEM Getting Started Guide*, available at:

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

Information about managing authentication is also available in the Postgres core documentation, available at:

<https://www.postgresql.org/docs/9.4/static/auth-pg-hba-conf.html>

SELinux Permissions

Before installing the PEM Server on a system that is running SELinux, you must set SELinux to `permissive` mode.

The following example works on Redhat Enterprise Linux, Fedora Core or CentOS distributions; use comparable commands that are compatible with your Linux distribution to set SELinux to `permissive` mode during installation and return it to `enforcing` mode when installation is complete.

Before installing the PEM Server, set SELinux to `permissive` mode with the command:

```
# setenforce Permissive
```

When the installation is complete, return SELinux to `enforcing` mode with the command:

```
# setenforce Enforcing
```

Firewall Restrictions

Please note that you must adjust your firewall to allow communication between PEM components.

RPM Installation of PEM Agent - Pre-requisites for RHEL or CentOS 7.x

Before installing the PEM agent on a CentOS 7.x host, you must ensure that the host contains the `wxBASE` and `epel-release` packages. To install the packages, open a command line, assume `root` privileges, and invoke the commands:

```
yum install epel-release
yum install wxBASE
```

Windows Permissions

If you are installing PEM components on Windows, you may be required to invoke the PEM installers with `Administrative` privileges. To invoke an installer using `Administrator` privileges, right click on the installer icon and select `Run as Administrator` from the context menu.

Supported Locales

At this time, the PEM server and web interface support a locale of `English (US)` `en_US` and use of a period (.) as a language separator character. Using an alternate locale, or a separator character other than a period may result in errors.

Windows Server IE Security

If you are using Internet Explorer on a Windows Server host to access monitored servers with the PEM web interface, you must disable Internet Explorer's Enhanced Security to take advantage of PEM functionality. If you do not wish to disable IE Enhanced Security, we recommend that you use an alternate browser (such as Chrome).

3 Installing Postgres Enterprise Manager™

You can use a graphical installer, StackBuilder Plus, or (on RHEL 7 only) an RPM package to add the Postgres Enterprise Manager™ server or agent to a host. Installers are available from the EnterpriseDB website at:

<http://www.enterprisedb.com/download-postgres-enterprise-manager>

The PEM server graphical installer installs and configures the PEM server, a PEM agent, and the software required to connect to the PEM web interface with your choice of browser. The PEM agent that is installed with the PEM server is capable of monitoring multiple servers that reside on the same host, or on remote hosts; please note that PEM functionality on servers monitored by a remote agent may be limited. For detailed information about using the PEM server graphical installer, see Section [3.1](#).

You can use an RPM package on a RHEL 7 host to install the PEM server; for detailed information about using the installer, see Section [3.2](#).

The PEM agent graphical installer installs and registers the PEM agent. For more information about using the PEM agent graphical installer, see Section [3.3.1](#).

You can also use an RPM package to install the PEM agent; for more information about the PEM agent RPM package, see Section [3.3.2](#).

3.1 Using a Graphical Installer to Install the PEM Server

At the heart of each PEM installation is the server. In a production environment, the server will typically be a dedicated machine, monitoring a large number of Postgres servers or a smaller number of busy servers.

The PEM server backend database may be an EnterpriseDB distribution of the PostgreSQL or Advanced Server database server, or an existing Postgres server installed from another source. The Postgres backing database server must be version 9.4 or later, and will contain a database named `pem`, which is used by the PEM server as a repository.

- If you would like to use an existing Postgres server to host the PEM server, the PEM server installer can create the `pem` database on the Postgres host. You must manually satisfy the software pre-requisites if you choose to use an existing server.

For more information about using an existing Postgres server to host the PEM server backend database, see [Section 3.1.3](#).

- If you do not wish to use an existing installation of Postgres as the PEM server host, the PEM server installer can install PostgreSQL, satisfy the server host's software pre-requisites, and create an instance (a PostgreSQL database cluster) that contains the `pem` database.

This is the simplest PEM server installation option.

PEM-HTTPD is made available for Postgres installations through the PEM server installer or the StackBuilder utility. If PEM-HTTPD is already installed on the host, the PEM server installer will review and update the existing installation if required. If the PEM server host does not contain an existing PEM-HTTPD installation, the PEM server installer will add it.

Before installing the PEM server, you must decide if you wish to run PostgreSQL and PEM-HTTPD on the same host or on separate hosts. If you intend to run the PostgreSQL database server and PEM-HTTPD on different hosts, then you must run the PEM server installer twice – once on each host, as detailed in [Section 3.1.2](#).

The PEM server installer will also install the software required to access the server via the PEM web interface (shown in Figure 3.1). You can access the web interface with a supported version of your browser of choice.

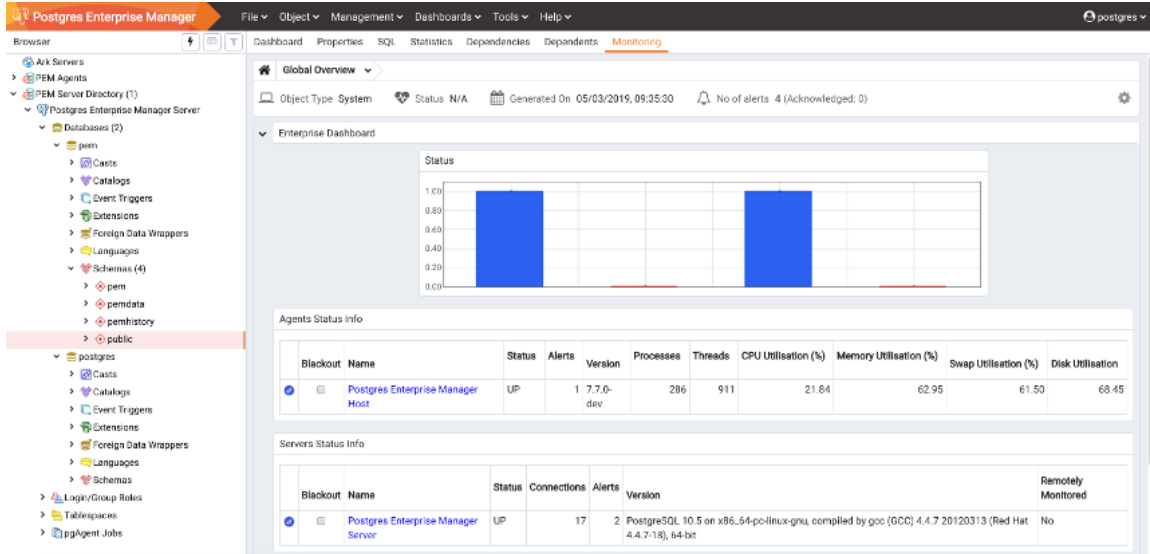


Figure 3.1 - The PEM web interface.

You can use the web interface to review information about objects that reside on monitored servers, manage databases and database objects that reside on monitored servers, or review statistical information gathered by the PEM server. The interface also provides access to clusters that reside on registered EDB Ark consoles.

3.1.1 Installing the PEM Server and PEM-HTTPD on the Same Host

The easiest PEM server installation configuration consists of a PEM backend database server (hosted on a PostgreSQL database installed with the PEM server installer) and a PEM-HTTPD service that reside on the same host. In this configuration, the PEM server installer will provide the pre-requisite software for the backend host and create a service script (on Linux) or register the service (on Windows).

To invoke the PEM server installer on Linux, assume superuser privileges and navigate into the directory that contains the installer. Then, invoke the installer with the command:

```
./pem_server-7.x.x-x-platform.ext
```

On a Windows system, right click the installer icon and select `Run as Administrator`. The installer displays a `Welcome` dialog (shown in Figure 3.2).

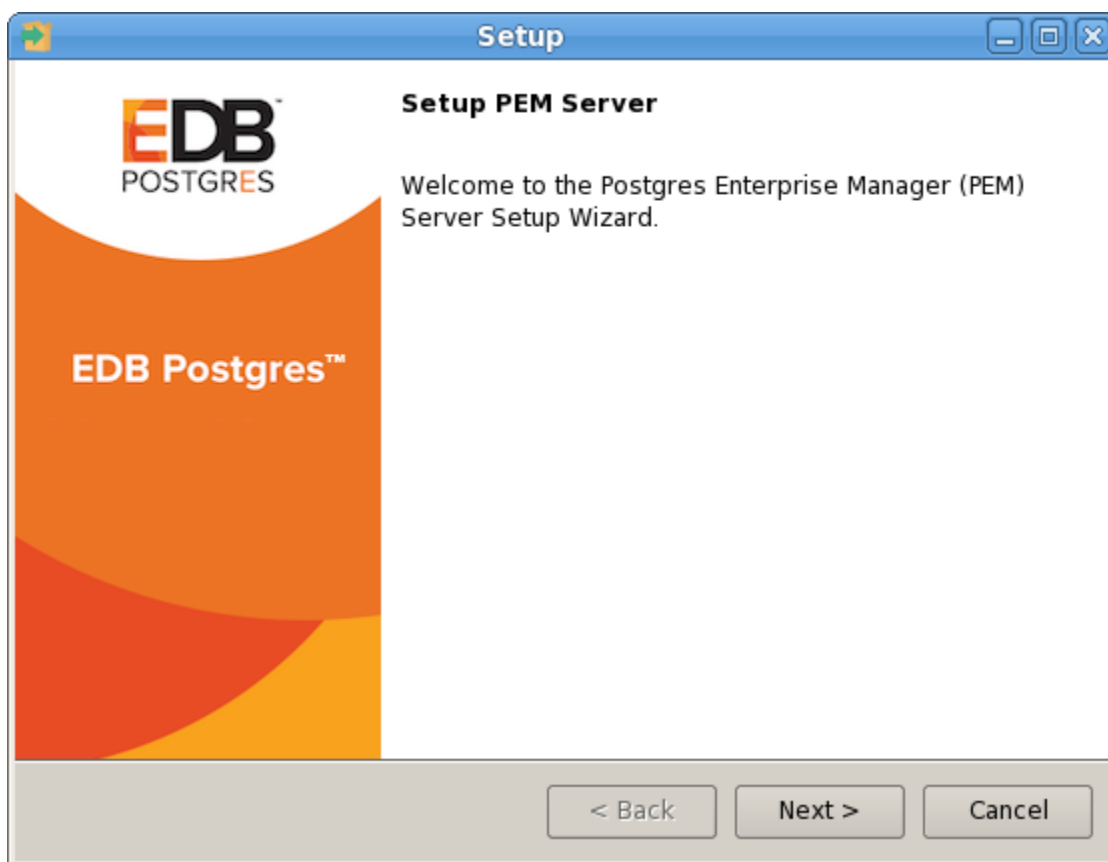


Figure 3.2 - The PEM Server Installer's Welcome dialog.

Click `Next` to continue to the `License Agreement` dialog.

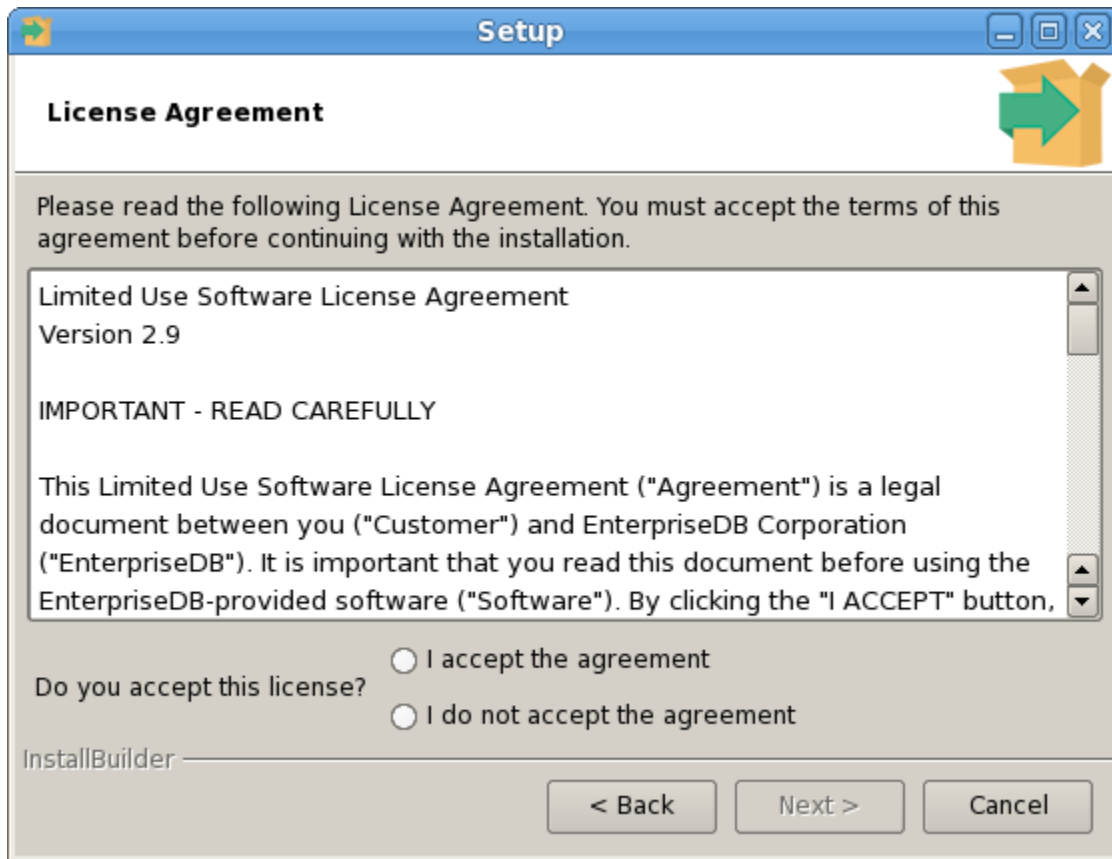


Figure 3.3 - The License Agreement.

Carefully review the license agreement (see Figure 3.3) before highlighting the appropriate radio button and accepting the agreement. Click `Next` to continue to the `Installation Directory` dialog.

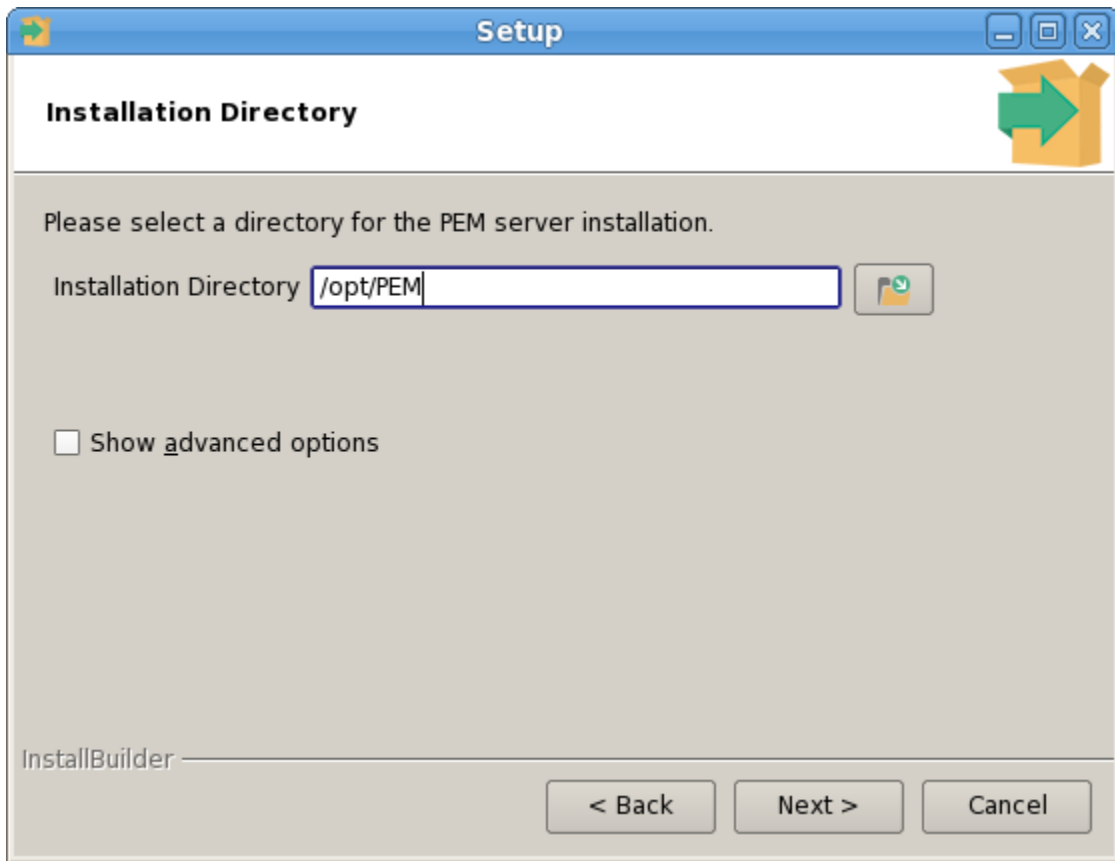


Figure 3.4 - Specify an installation directory.

Use the `Installation Directory` dialog (shown in Figure 3.4) to specify the location of the PEM server:

- By default, the PEM server is installed in the `/opt/PEM` directory on Linux or in `C:\Program Files\PEM` on Windows. Accept the default location, or use the `Installation Directory` button to open a browser dialog and select the directory in which the PEM server will be installed.
- Use the `Show advanced options` check box to instruct the installer to open the `Advanced options` dialog.

Use the `Advanced options` dialog when installing the Postgres database server and the PEM-HTTPD on different hosts, or if you wish the PEM server to reside on an existing Postgres server installation.

To install the PostgreSQL server packaged with the installer and PEM-HTTPD on the same host, leave the `Show advanced options` box unchecked and click `Next`.

The PEM server installer will perform a pre-installation check for PEM-HTTPD, LanguagePack, and PostgreSQL 10. If the installer does not locate these packages, it will inform you in the `Dependency missing` dialog (shown in Figure 3.5).

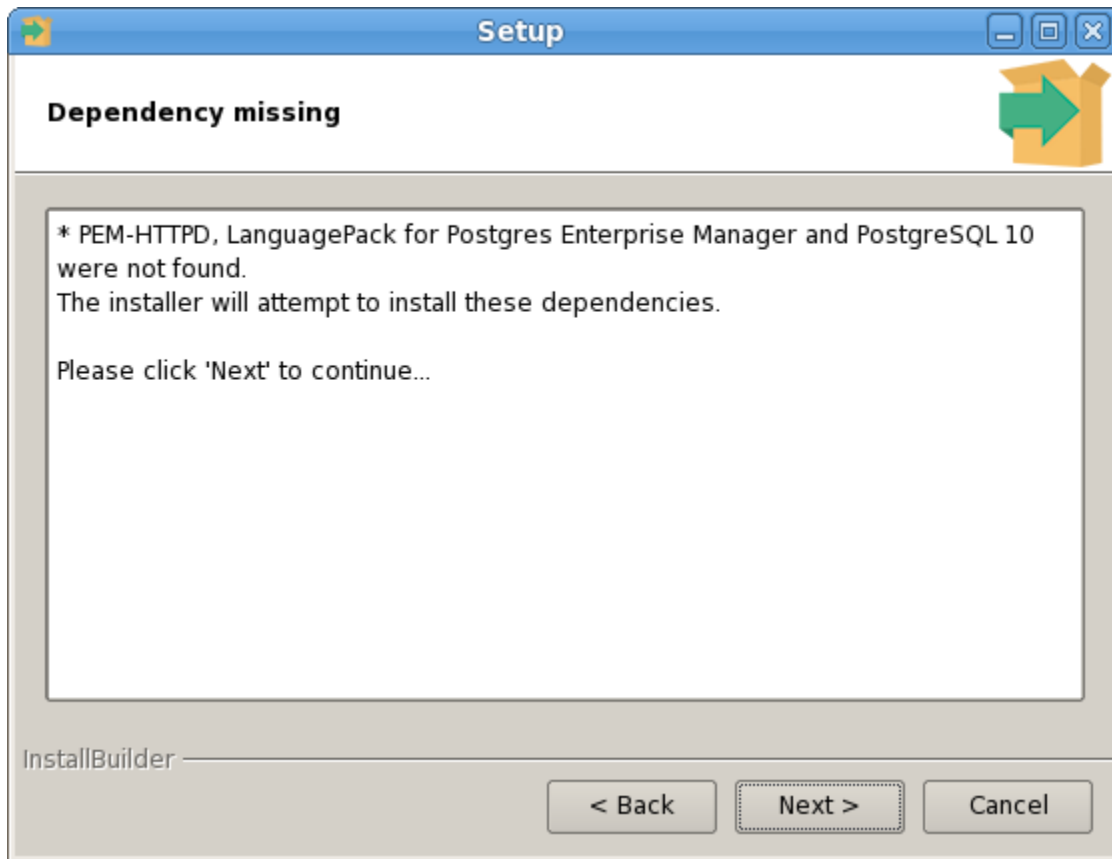


Figure 3.5 - The installer checks for pre-requisites.

If the dependencies are missing, the PEM server installer will launch the respective installation wizards; follow the onscreen directions presented by the installation wizards for each package.

After installing any missing dependencies, the installation process continues by displaying the `Database Server Installation Details` dialog.

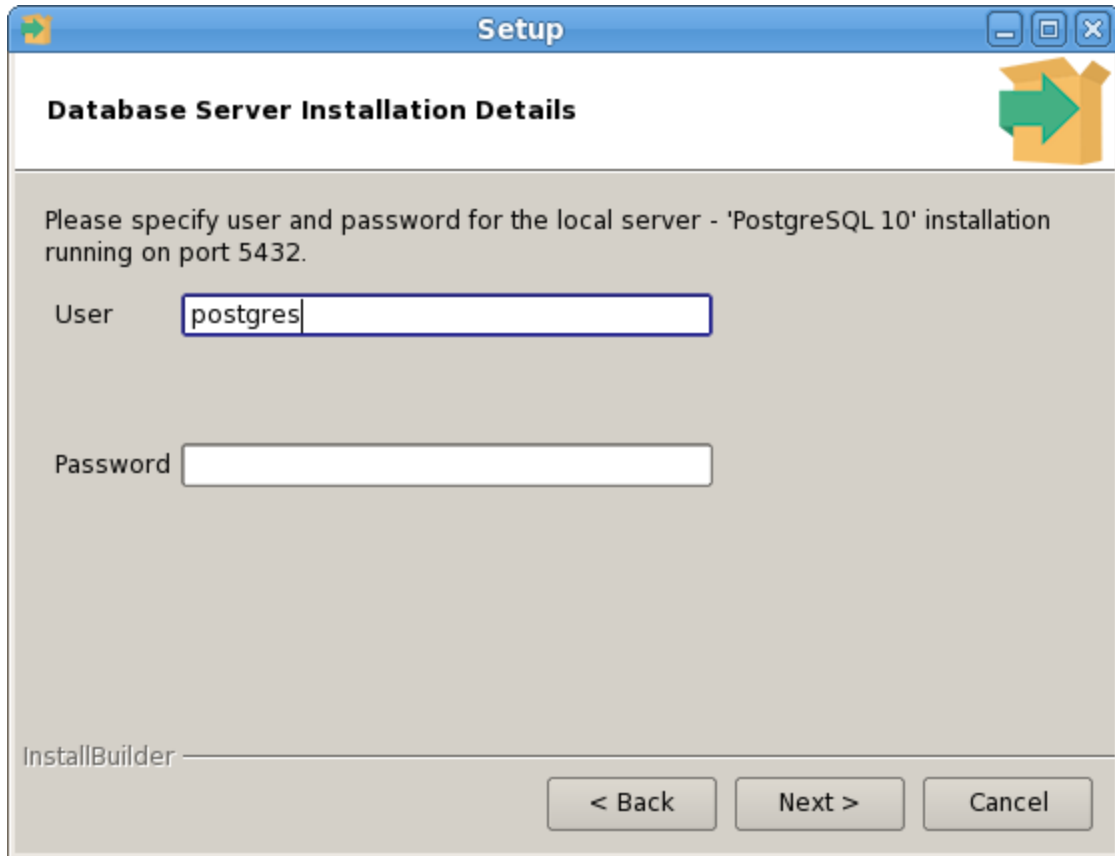


Figure 3.6 - Enter the name and password of the PostgreSQL superuser.

The information provided on the Database Server Installation Details dialog (see Figure 3.6) enables the installer to connect to the PostgreSQL server. Provide the User name and Password of a database superuser. After supplying the requested information, click Next to continue to the Network Details dialog.

After providing the name and password of the Postgres database superuser, you may be prompted for the password to the user account under which the PEM agent will run. If prompted, provide the password, and press Next to continue.

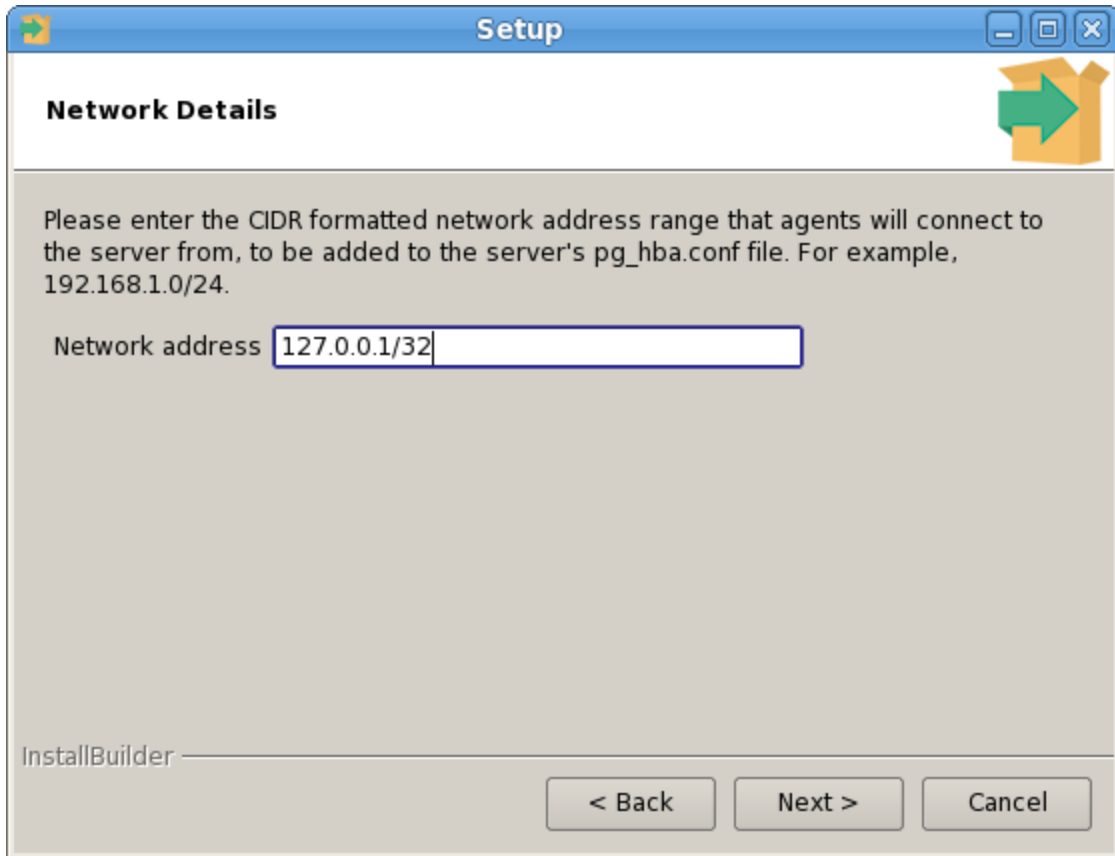


Figure 3.7 - Supply the network address from which the agent will connect.

Use the `Network Details` dialog (see Figure 3.7) to specify the CIDR-style network address from which the PEM agents will connect to the server (the *client-side* address).

You may specify the address of a network host, or a network address range. For example, if you wish to monitor database servers with the addresses `192.168.10.23`, `192.168.10.76` and `192.168.10.184`, enter `192.168.10.0/24` to allow connections with hosts in that network.

The specified address will be added to the server's `pg_hba.conf` file. You can specify additional network addresses by manually adding entries to the `pg_hba.conf` file on the PostgreSQL server if required, using the initial entry as a template.

When you've added the `Network address`, click `Next` to continue to the `Agent Details` dialog.

The PEM server installer will install a PEM agent on the host on which the server resides, to monitor the server and provide alert processing and garbage collection services. A certificate will also be installed in the location specified in the Agent certificate path field (see Figure 3.8).

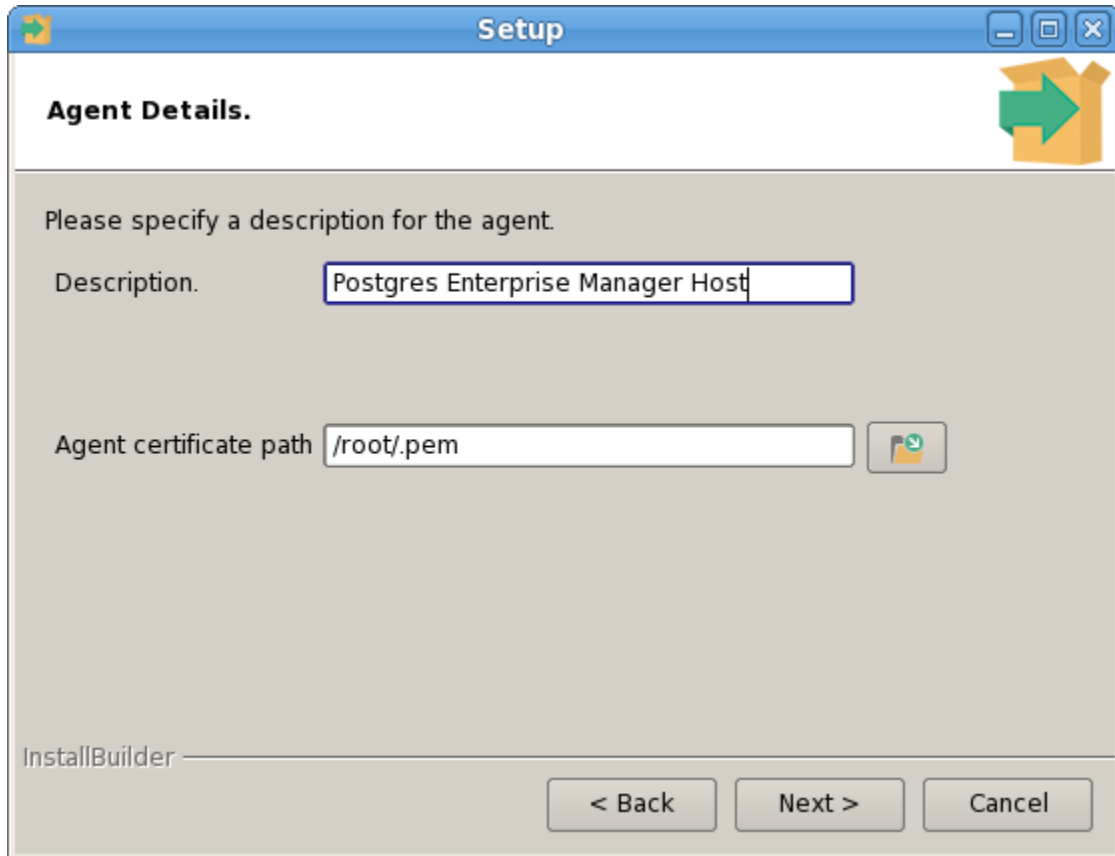


Figure 3.8 - Provide a Description for the agent that resides on the server.

Enter an alternate description or select an alternate agent certificate path for the PEM agent, or accept the defaults. Click `Next` to continue to the `Ready to Install` dialog.

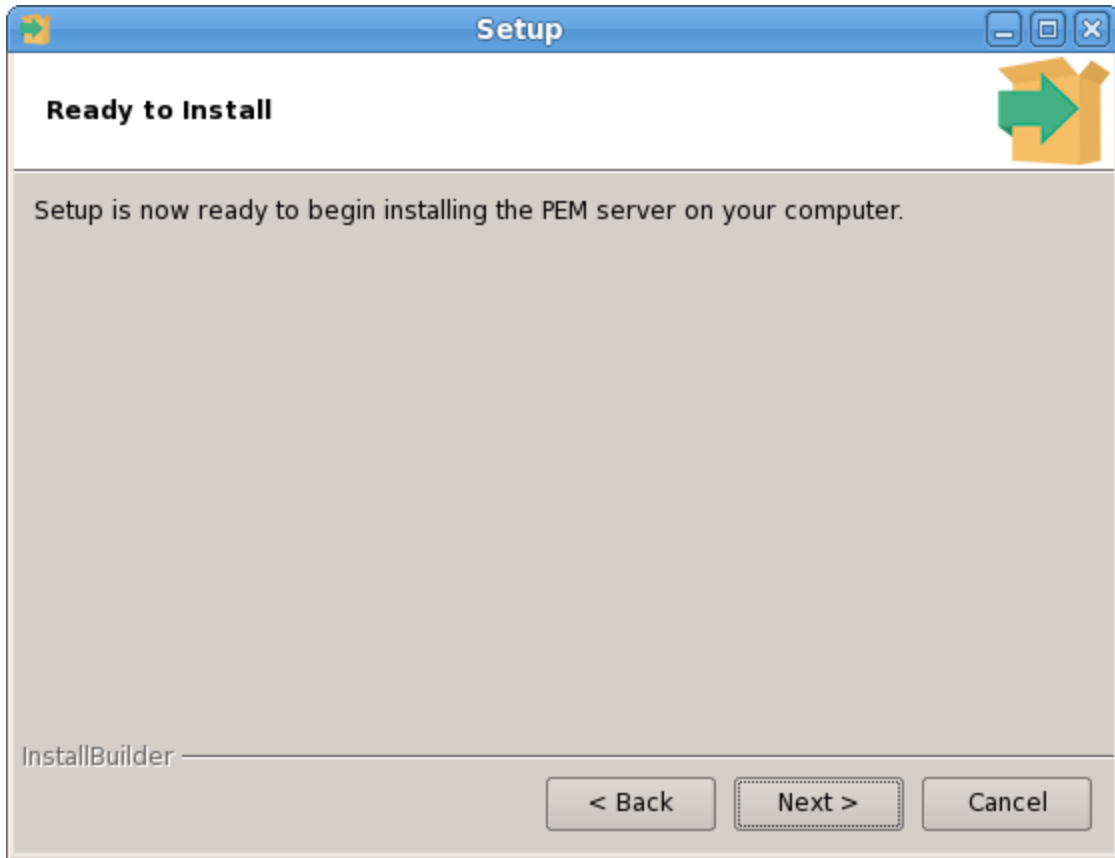


Figure 3.9 - The installation is ready to begin.

The wizard is now ready to install the PEM server (see Figure 3.9).

Click `Back` to modify any of the options previously selected, or `Next` to continue with the installation.

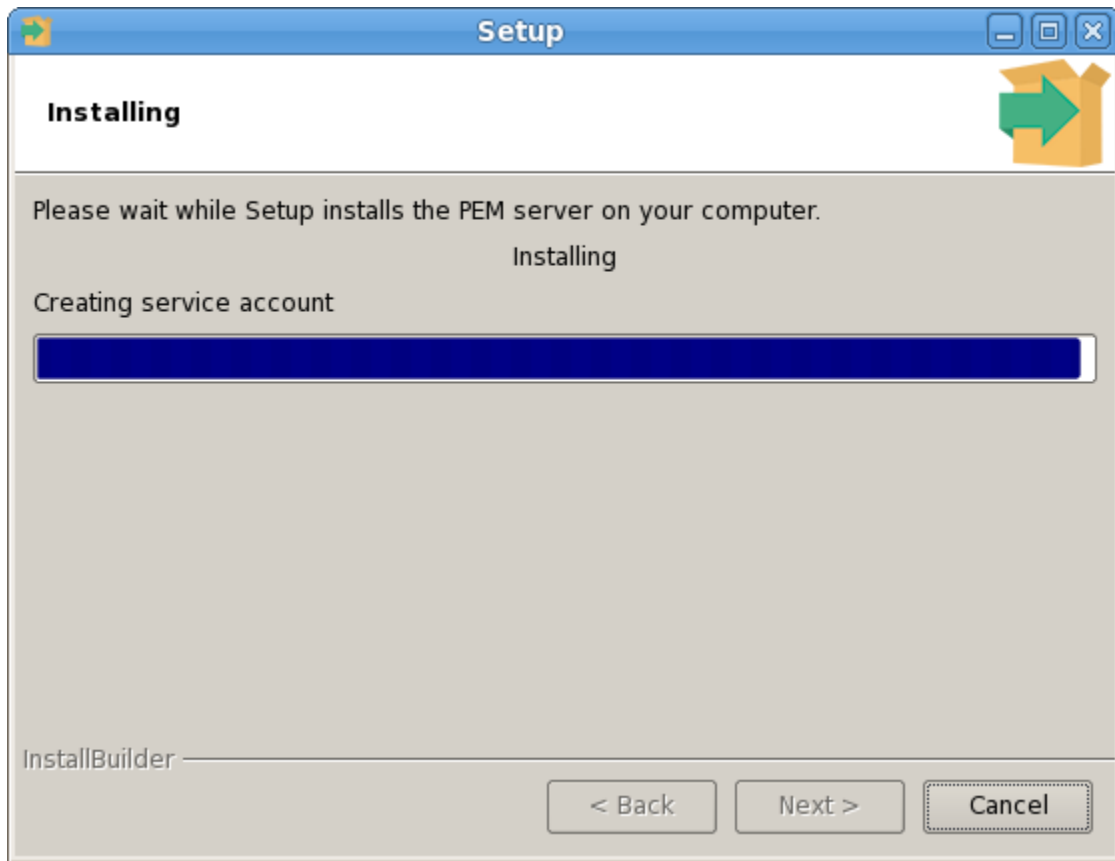


Figure 3.10 - The installation in progress.

During the installation process, the installer will copy files to the system, and set up the database and web services required to run PEM. When the installation completes, a popup dialog opens (shown in Figure 3.11) confirming that the webservice has been configured, and is listening on port 8443, and that the `pem` database has been created and configured.

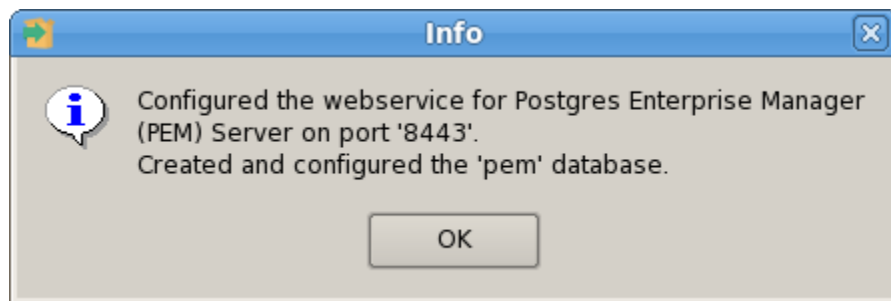


Figure 3.11 - A popup confirms the installation details.

Click **OK** to acknowledge that the webservice has been configured, and that the `pem` database has been created, and continue to the `Completed...` dialog (shown in Figure 3.12).

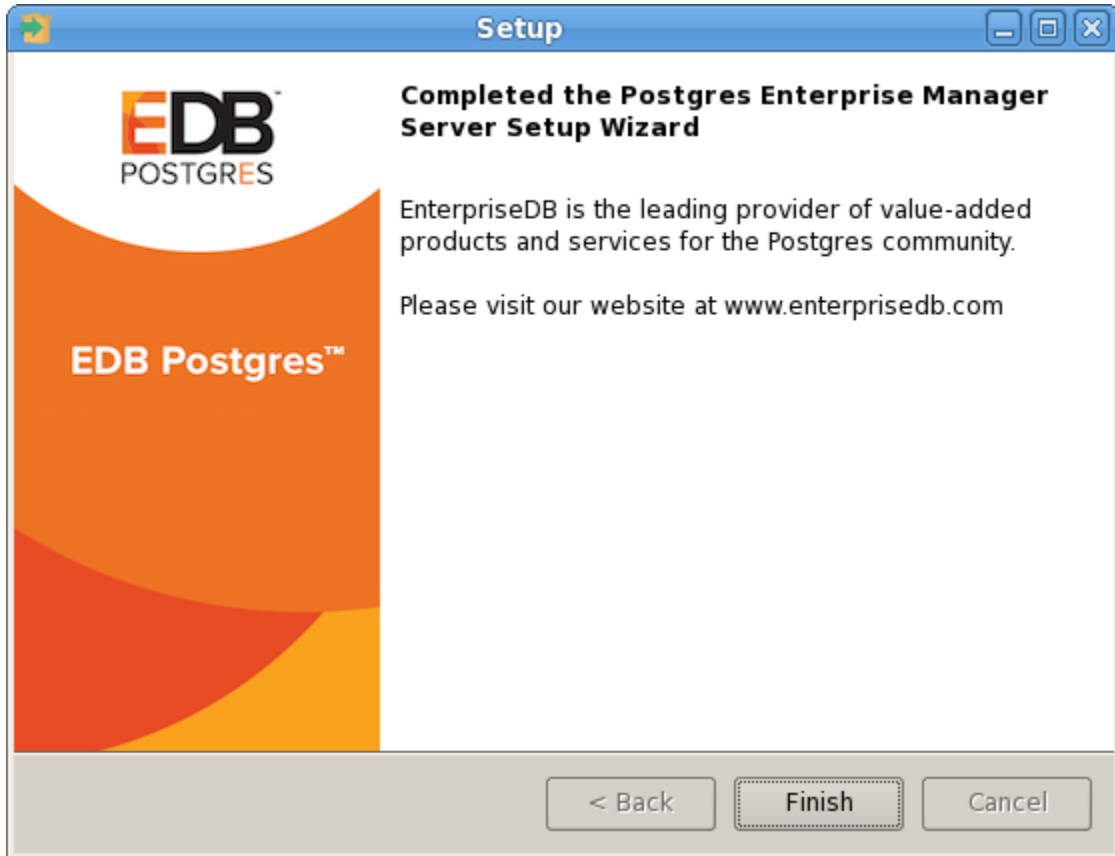


Figure 3.12 - The PEM server installation is complete.

3.1.2 Installing the PEM Server and PEM-HTTPD on Separate Hosts

To use separate hosts for the PEM server backend database and PEM-HTTPD you must:

1. Invoke the PEM server installer on the host of the Postgres server that will contain the `pem` database. During the installation, select the `Database` option on the `Advanced options` dialog, and provide connection information for the Postgres server.
2. Modify the `pg_hba.conf` file of the Postgres installation on which the PEM server (and `pem` database) resides, allowing connections from the host of the PEM-HTTPD server.
3. Invoke the PEM server installer on the host of the PEM-HTTPD server, selecting the `Web Services` option on the `Installation Type` dialog.

To invoke the PEM server installer on Linux, assume superuser privileges and navigate into the directory that contains the installer. Then, invoke the installer with the command:

```
./pem_server-7.x.x-x-platform.ext
```

On a Windows system, right click the installer icon and select `Run as Administrator`. The installer displays a `Welcome` dialog (shown in Figure 3.13).



Figure 3.13 - The PEM Server Installer's Welcome dialog.

Click **Next** to continue to the **License Agreement** dialog.

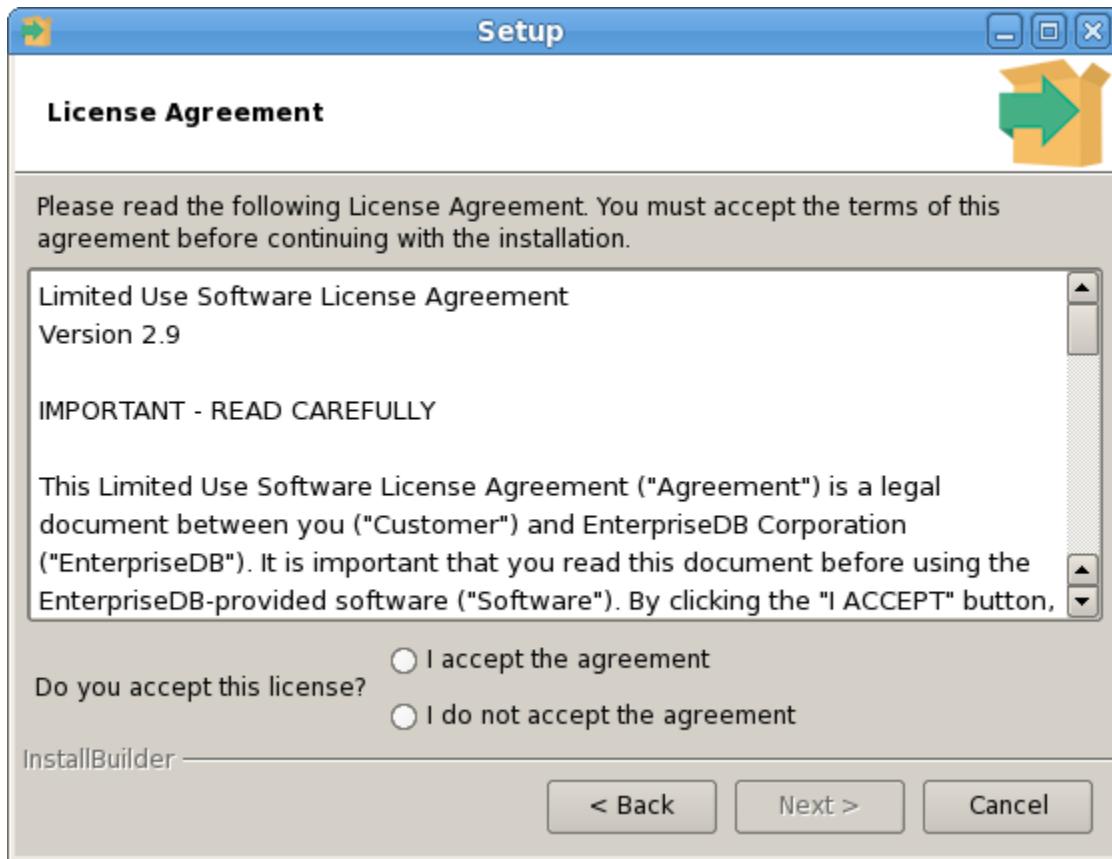


Figure 3.14 - The License Agreement.

Carefully review the license agreement (see Figure 3.14) before highlighting the appropriate radio button and accepting the agreement; click `Next` to continue.

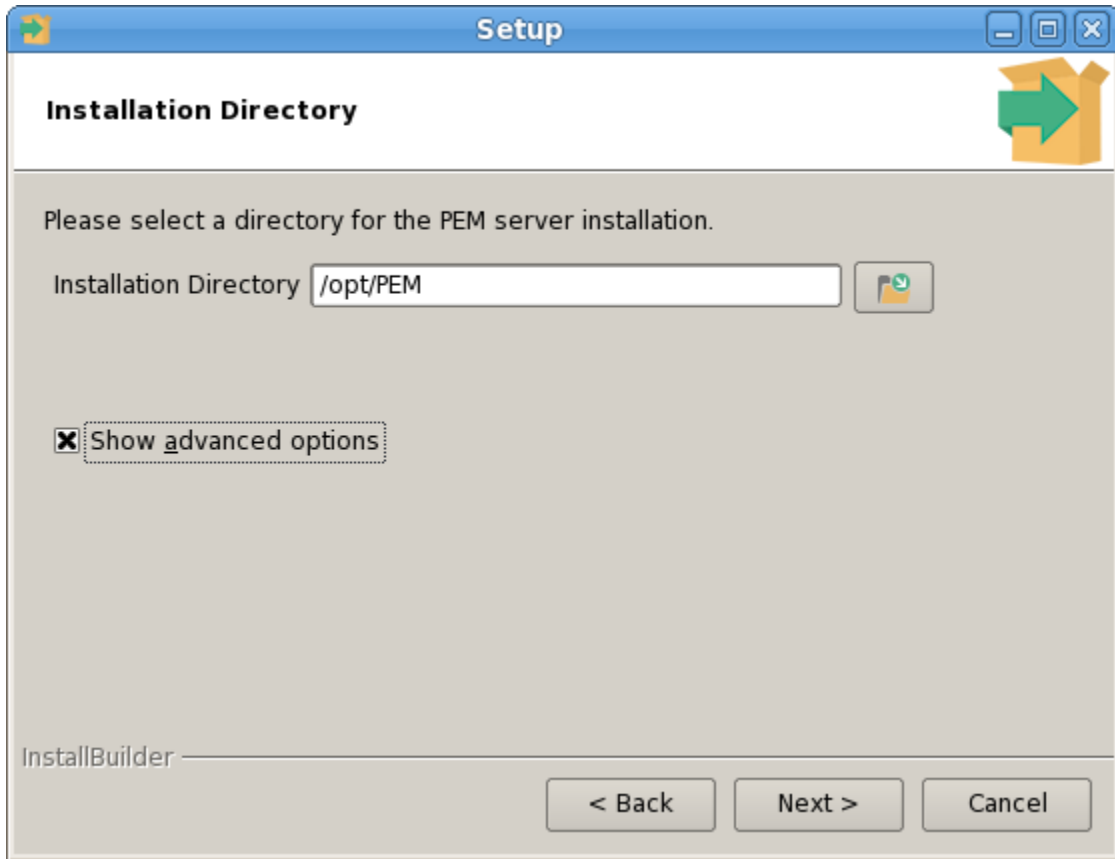


Figure 3.15 - The Installation Directory dialog.

Use fields on the `Installation Directory` dialog (see Figure 3.15) to specify the directory in which the PEM server will reside, and to access the `Advanced options` dialog:

- By default, the PEM server is installed in the `/opt/PEM` directory on Linux or in `C:\Program Files\PEM` on Windows. Accept the default location, or use the `Installation Directory` field to open a browser dialog and select the directory in which the PEM server will be installed.
- To install the PEM server and PEM-HTTPD on separate hosts, you must use the `Advanced options` dialog to specify the installation type (`Web Services` or `Database`). Select the `Show advanced options` check box to instruct the installer to include the `Advanced options` dialog in the installation process.

Click `Next` to continue to the `Advanced Options` dialog.

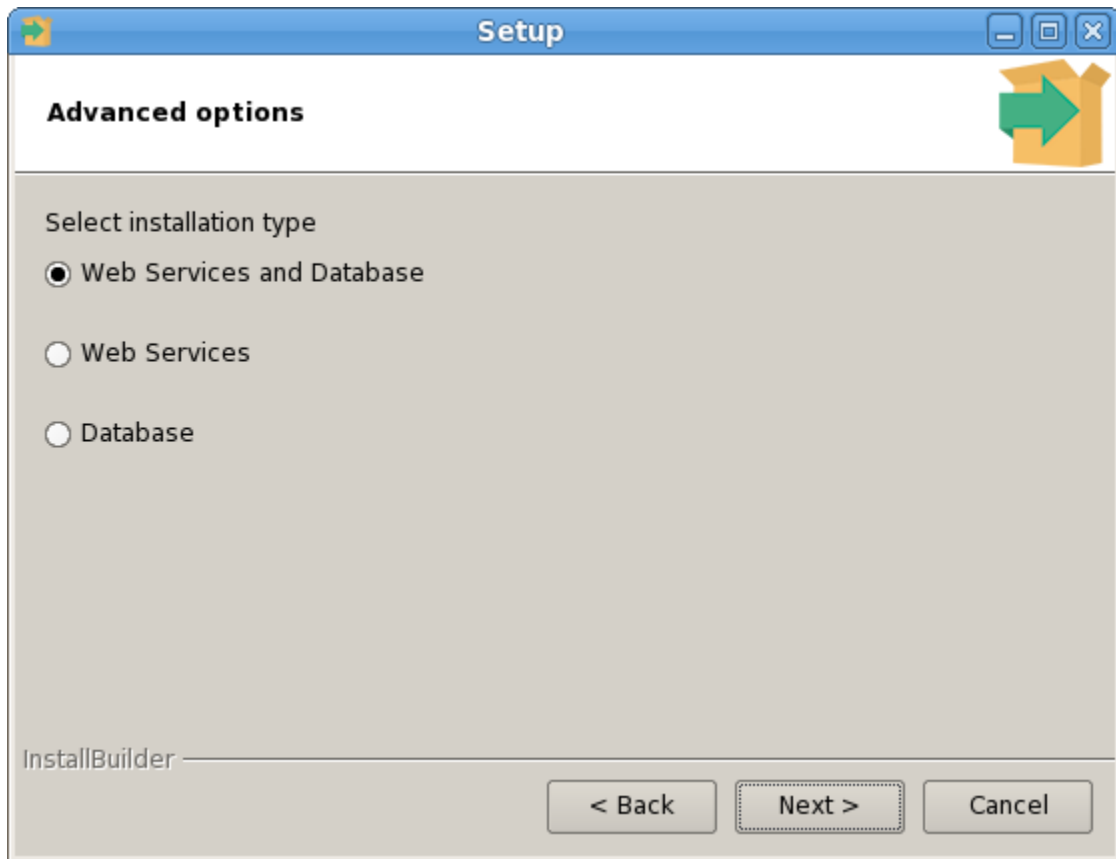


Figure 3.16 - The Advanced Options dialog.

Use the radio buttons on the `Advanced options` dialog (see Figure 3.16) to specify the component or components that you would like to install:

- Select `Web Services and Database` to indicate that the Postgres server and PEM-HTTPD will both reside on the current host. If you select the `Web Services and Database` option, the PEM server installer will allow you to specify which Postgres server you wish to use for the PEM server before checking for a PEM-HTTPD installation.
- Select `Web Services` to install PEM-HTTPD on the current host, while using a Postgres database server that resides on another host to host the PEM server and pem database.

Please note: you must complete the PEM server installation process on the host of the PEM server (and pem backend database), selecting `Database` on the `Advanced options` dialog, and modifying the connection properties of the `pg_hba.conf` file on the PEM server before using the `Web Services` option to install PEM-HTTPD.

This option invokes the installation steps documented in [Section 3.1.2.2](#), *Installing Web Services*.

- Select `Database` to use an existing Postgres server (version 9.4 or greater), or to install only the database server that is distributed with the PEM server installer.

This option invokes the installation steps documented in [Section 3.1.2.1](#), *Specifying a Database Host*.

After selecting an installation option, click `Next` to continue.

3.1.2.1 Specifying a Database Host

Selecting the `Database` option on the `Advanced options` dialog (Figure 3.17) allows you to specify connection information for the host on which the PEM server backend database (named `pem`) will reside.

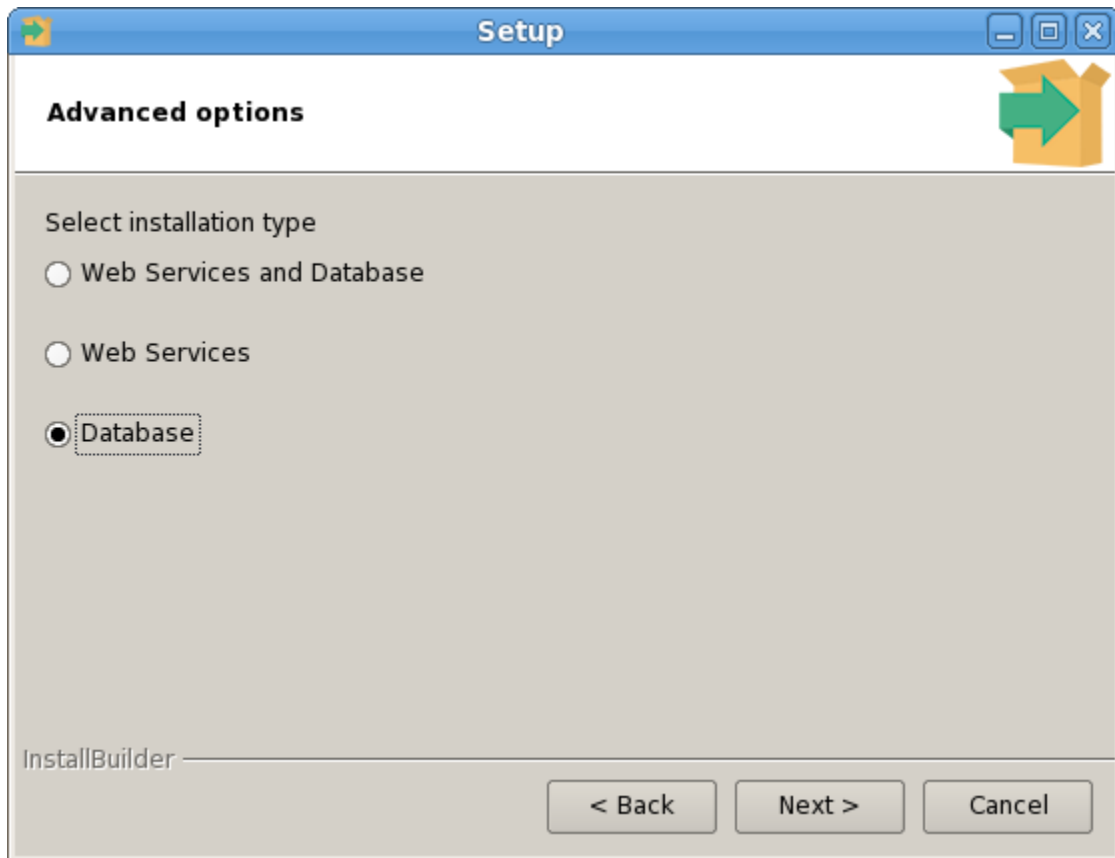


Figure 3.17 - The Advanced options dialog.

Click `Next` to continue to the `Database Server Selection` dialog.

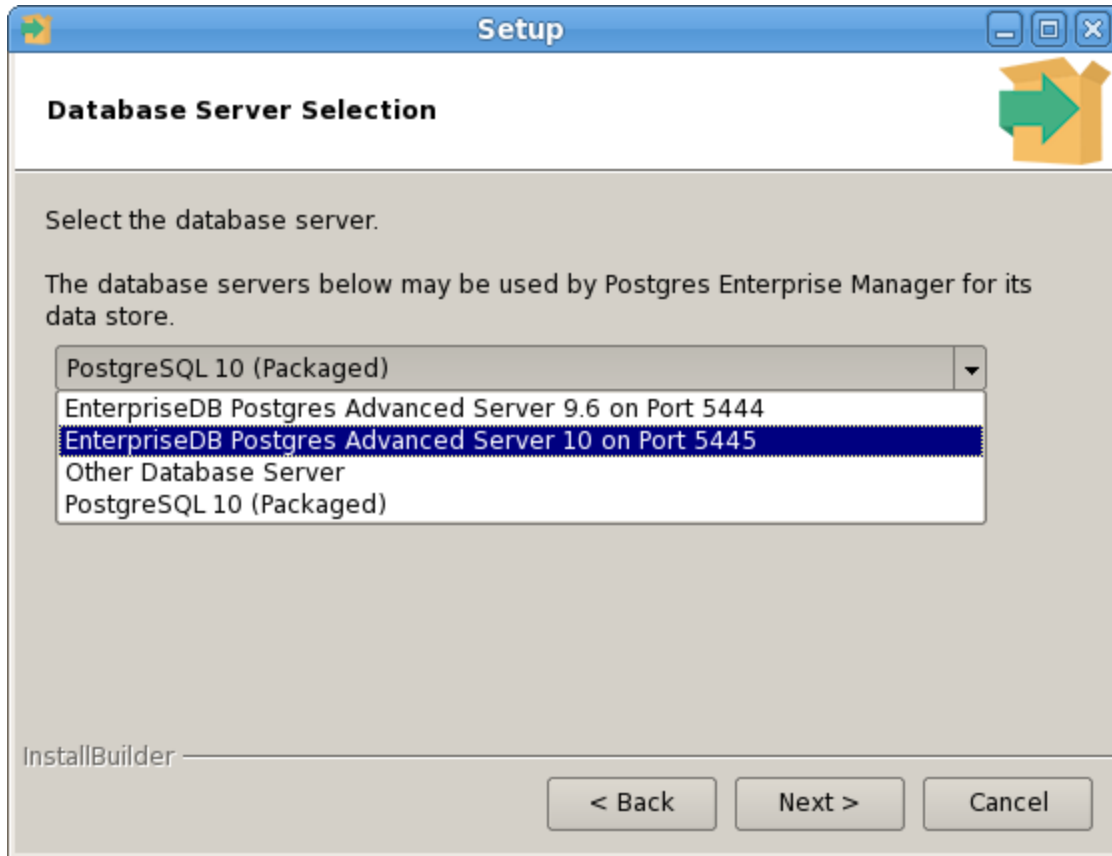


Figure 3.18 - Selecting a database server.

Use the drop-down listbox on the Database Server Selection dialog (see Figure 3.18) to select a host for the PEM server backend database. You can:

- Select a host from existing Postgres installations that reside on the current host. Please note that you may be required to add the `sslutils` package to your installation.
- Select the PostgreSQL 9.6 option to install the Postgres server that is distributed with the PEM server installer. If you decide to use the version of PostgreSQL that is bundled with the PEM server installer, the EnterpriseDB one-click PostgreSQL installer will open and walk you through the installation.
- Select Other Database Server to specify connection information for a Postgres server that was not installed using a one-click graphical installer from EnterpriseDB. For information about the software pre-requisites for the PEM server database host, please see [Section 3.1.3.1, Preparing the Postgres Server](#).

Click `Next` to continue.

If the PEM server will reside on an existing Postgres server, the Database Server Installation Details dialog shown in Figure 3.19 opens.

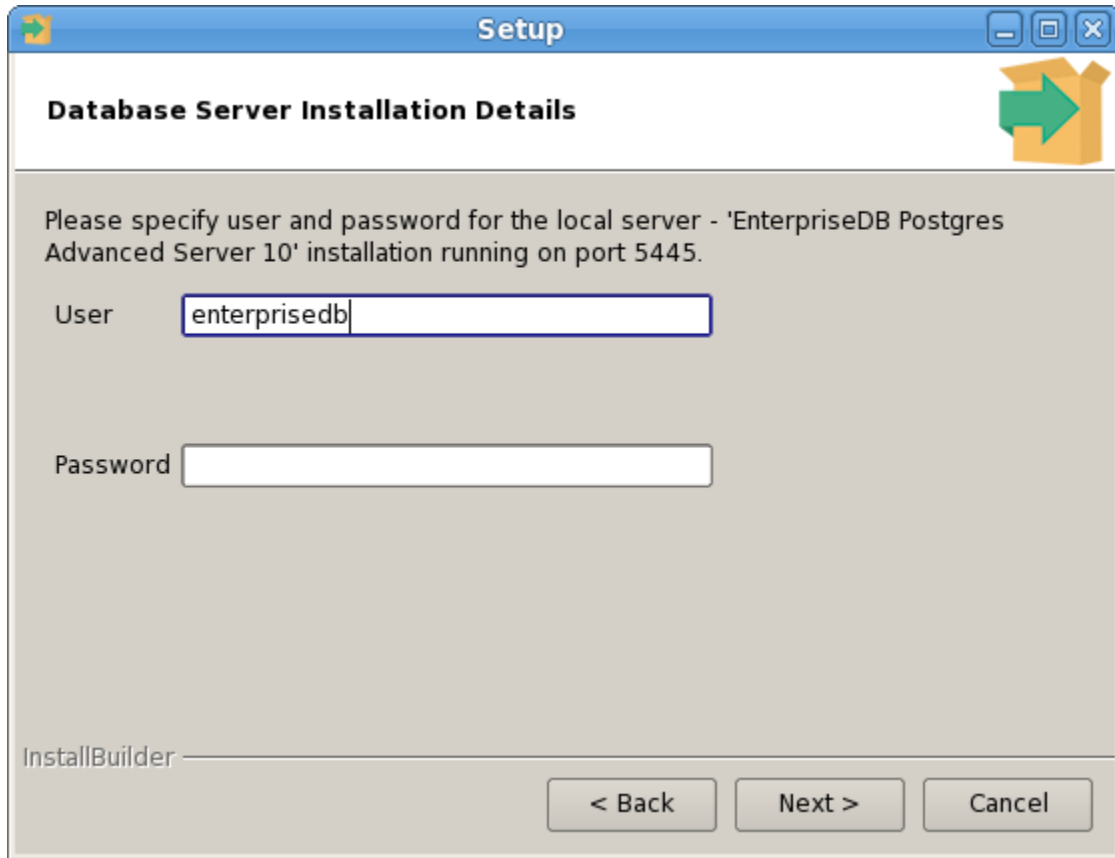


Figure 3.19 - The Database Server Installation Details dialog.

The information required on the Database Server Installation Details dialog may vary; the PEM server installer will ask you to provide only the information about the selected installation that it cannot locate:

- Specify the name of a Postgres database superuser in the `User` field.
- Specify the password associated with that user in the `Password` field.

Click `Next` to continue.

If prompted, provide the system password for the service account under which the PEM agent will run, and click `Next` to continue to the `Network Details` dialog.

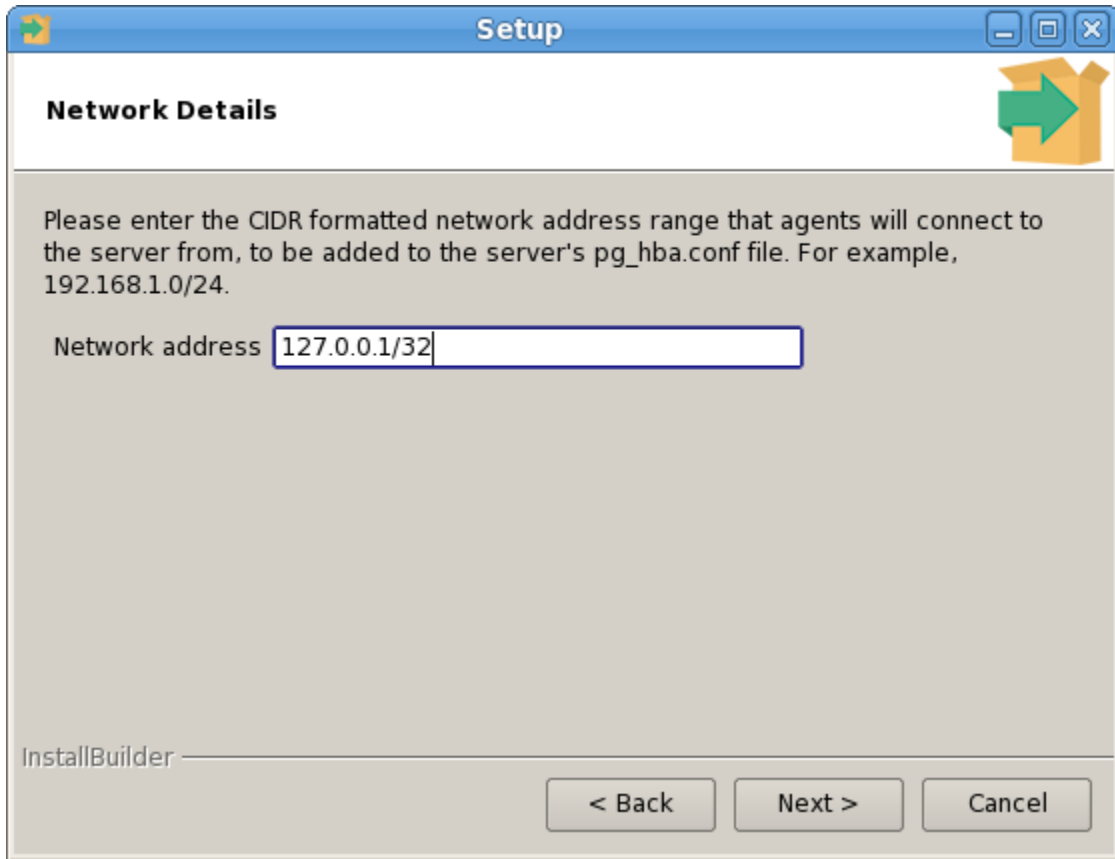


Figure 3.20 - Supply the network address from which the agent will connect.

Use the `Network Details` dialog (see Figure 3.20) to specify the CIDR-style network address from which PEM agents will connect to the server (the *client-side* address). The specified address will be added to the server's `pg_hba.conf` file.

Please note: you can specify additional network addresses by manually adding entries to the `pg_hba.conf` file on the PostgreSQL server.

Accept the default (specifying the localhost), or specify a `Network address range`, and click `Next` to continue to the `Agent Details` dialog.

The PEM server installer will install a PEM agent on the host on which the server resides, to monitor the server and provide alert processing and garbage collection services. A certificate will also be installed in the location specified in the `Agent certificate path` field (see Figure 3.21).

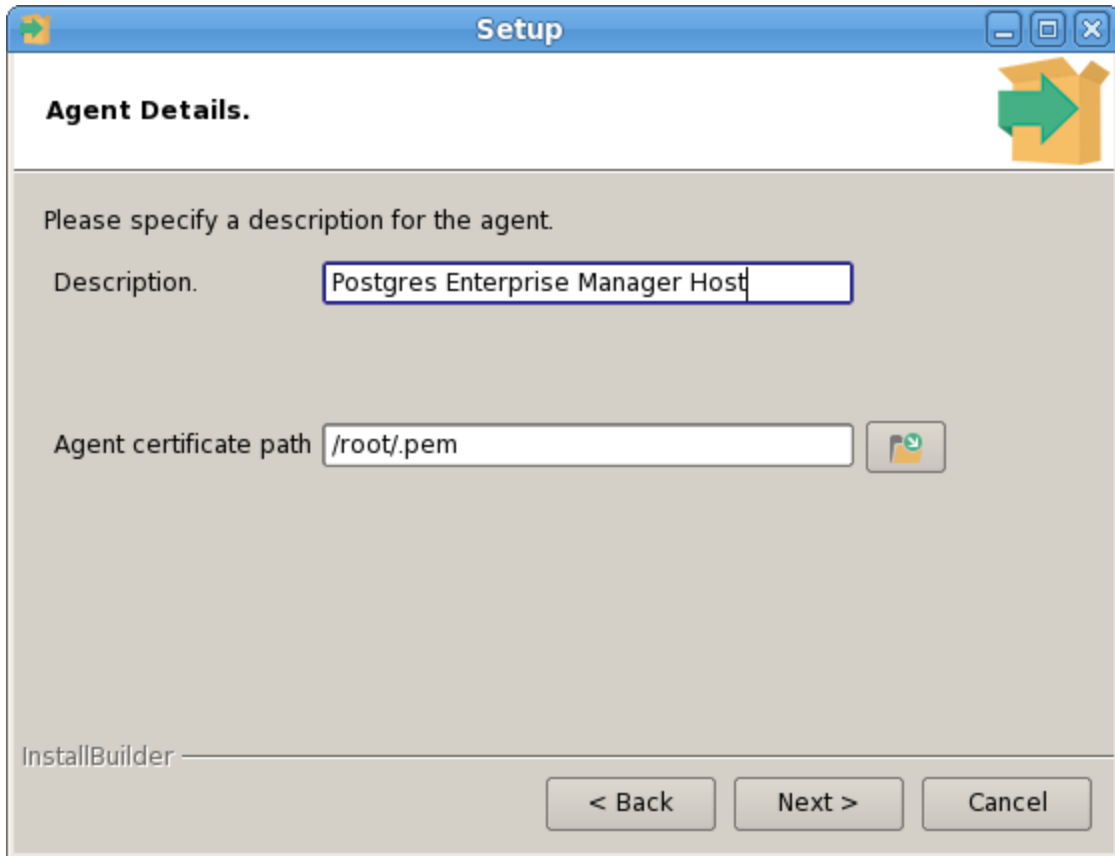


Figure 3.21 - Provide a Description for the agent that resides on the server.

You can enter an alternate description or an alternate agent certificate path for the PEM agent, or accept the defaults. Click `Next` to continue.

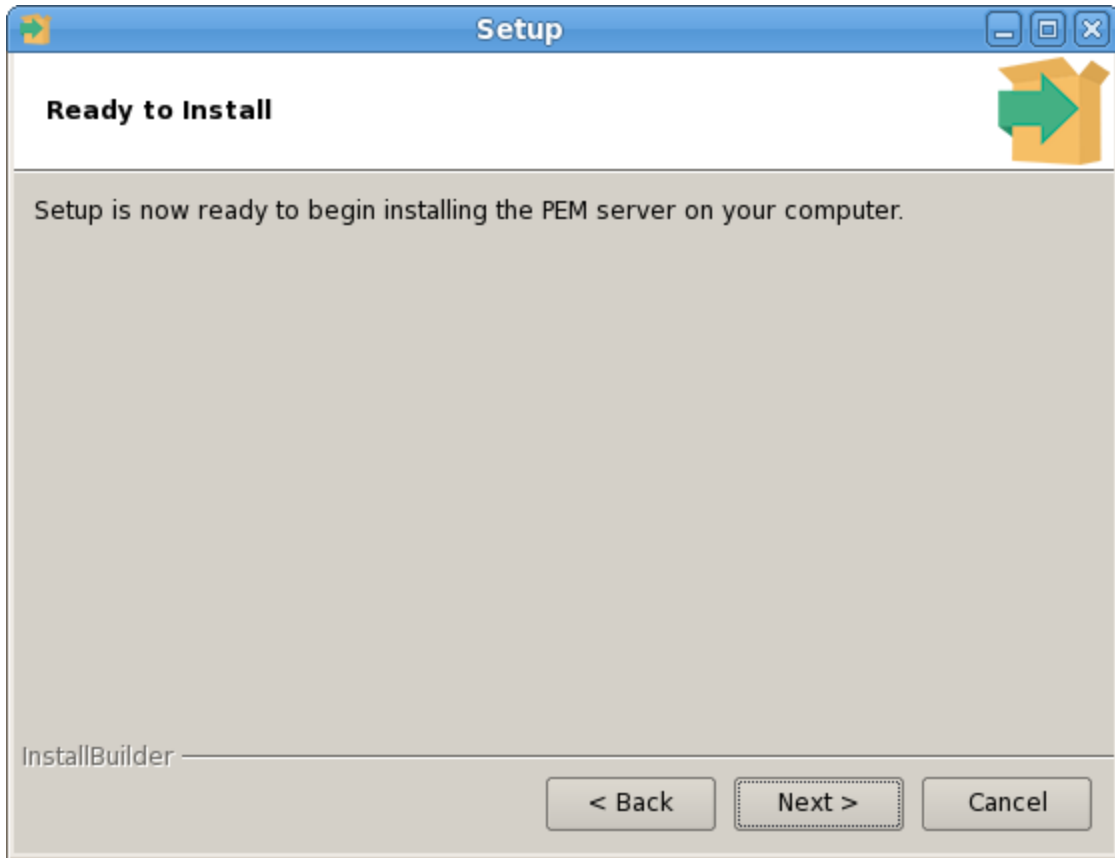


Figure 3.22 - The installation is ready to begin.

The wizard is now ready to install the PEM server (see Figure 3.22). Click `Back` to modify any of the options previously selected, or `Next` to proceed with the installation.

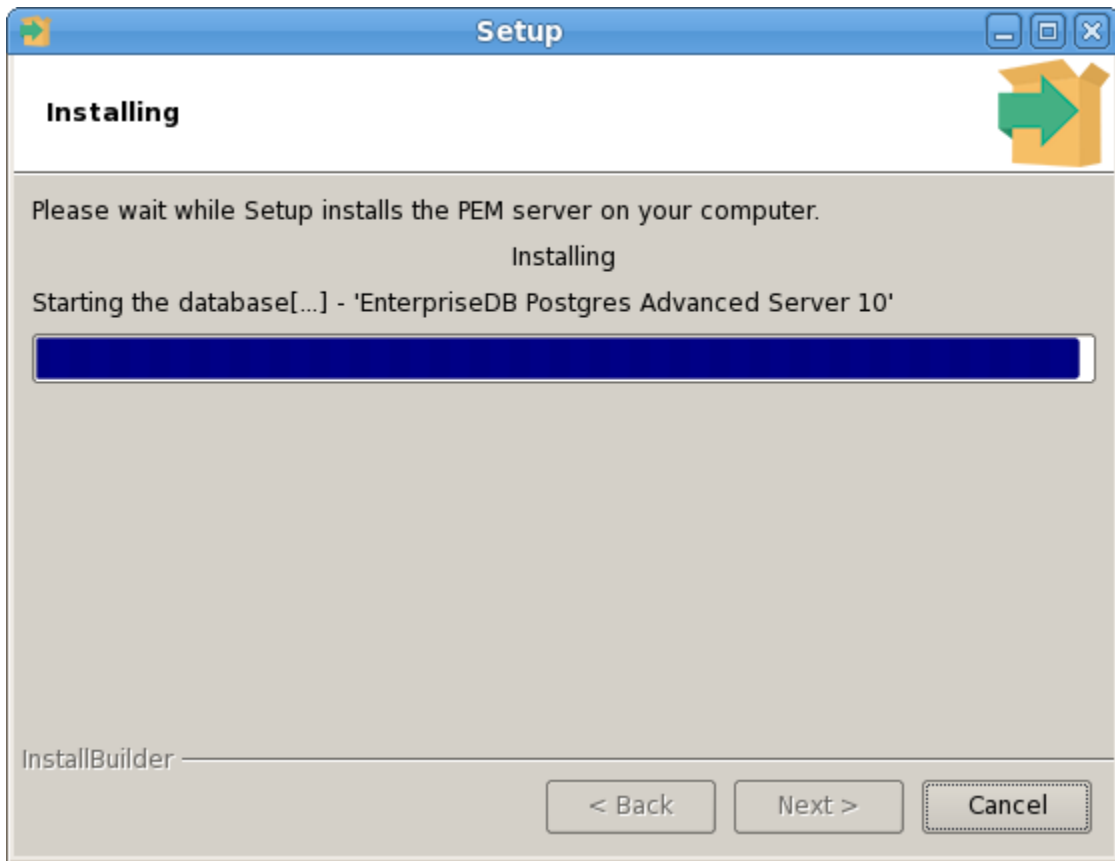


Figure 3.23 - The installation in progress.

During the installation process, the installer will copy files to the system, and set up the PEM server's backend database. A popup dialog opens (shown in Figure 3.24) confirming that the `pem` database has been created and configured.

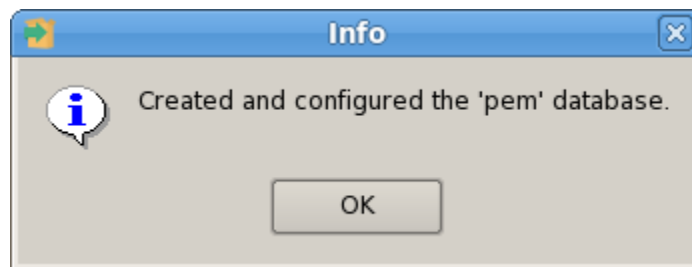


Figure 3.24 - Click OK to confirm that the pem database has been created.

Click `OK` to acknowledge that the `pem` database has been created, and continue to the `Completed...` dialog.

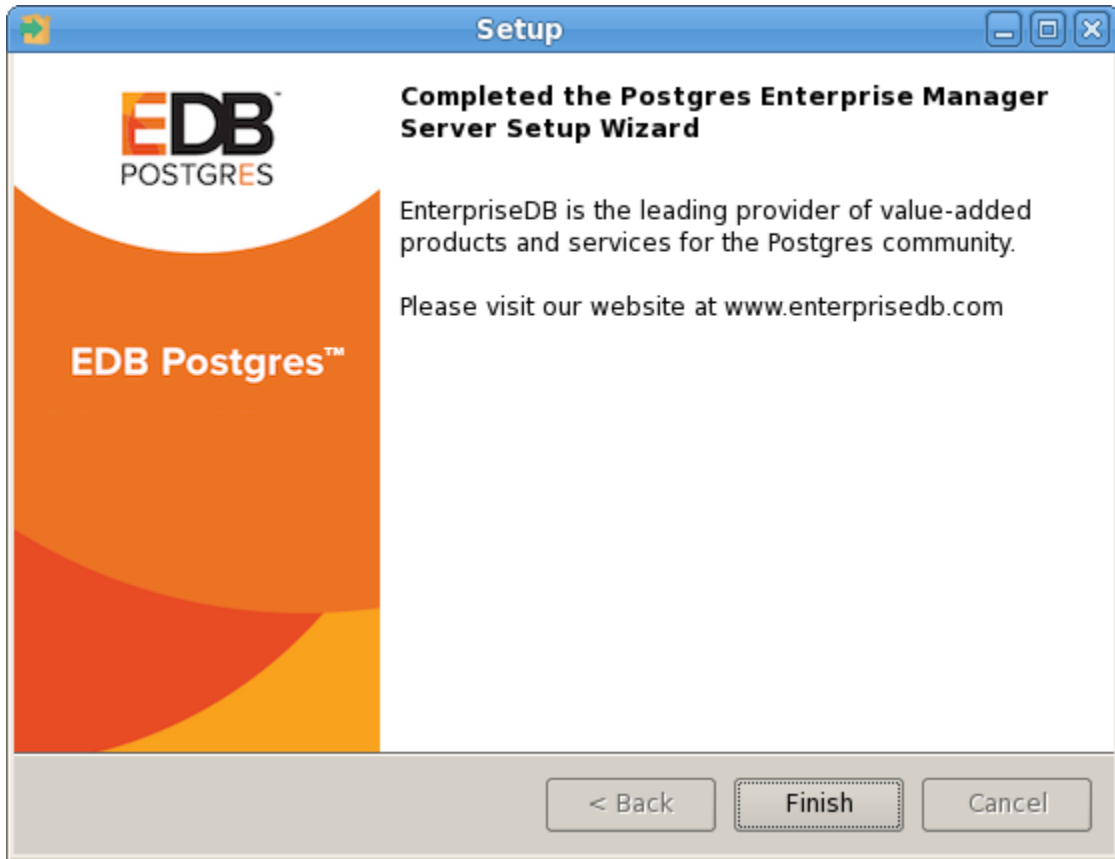


Figure 3.25 - The PEM server installation is complete.

When the database portion of the PEM server installation is complete (see Figure 3.25), you can invoke the PEM server on another host to install (or upgrade) PEM-HTTPD.

3.1.2.2 Installing Web Services

Selecting the `Web Services` radio button on the `Advanced options` dialog (Figure 3.26) instructs the PEM server installer to either install PEM-HTTPD on the current host or update an existing PEM-HTTPD installation.

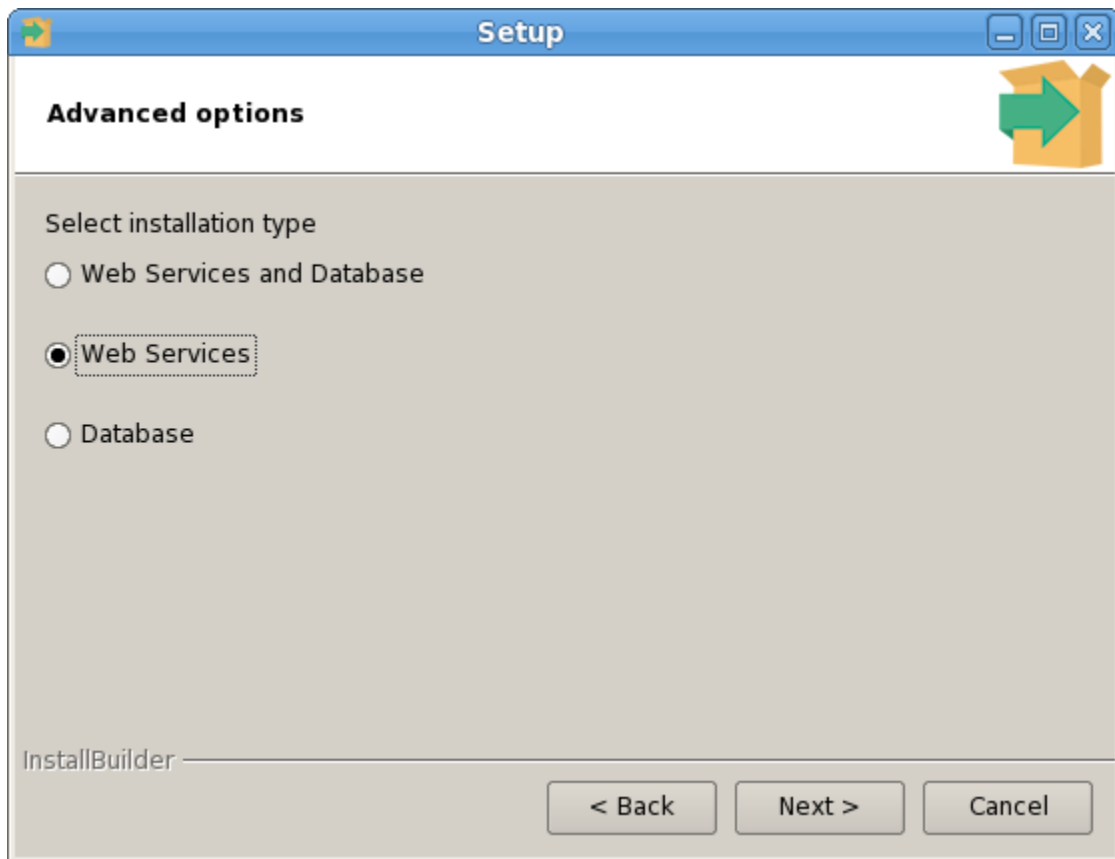


Figure 3.26 - Selecting the Web Services option on the Advanced options dialog.

Please note that the current host may not be the host of the PEM backing database. Before selecting this option, you must have:

- Completed an installation of the PEM server installer on a host system, during which you specified a backing database for the PEM server.
- Modified the `pg_hba.conf` file on the PEM server database host to allow connections from the PEM-HTTPD host, and restarted the database server.

When you select the `Web Services` option and click `Next`, the PEM server installer will check the current host for existing PEM-HTTPD and LanguagePack installations.

If the installer does not locate the components, the installer will inform you that one or more dependencies are missing (see Figure 3.27).

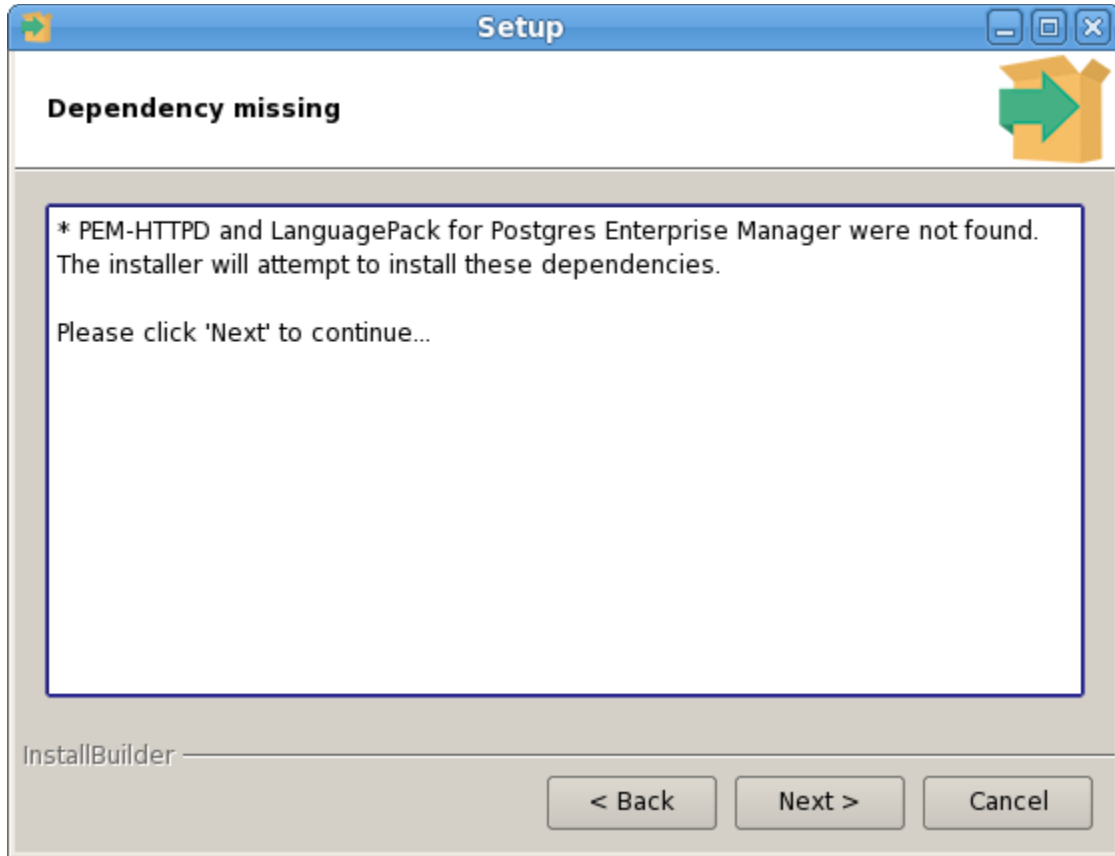


Figure 3.27 –The installer has detected missing dependencies.

Click **Next** to instruct the server to instruct the server to install LanguagePack (see Figure 3.28).

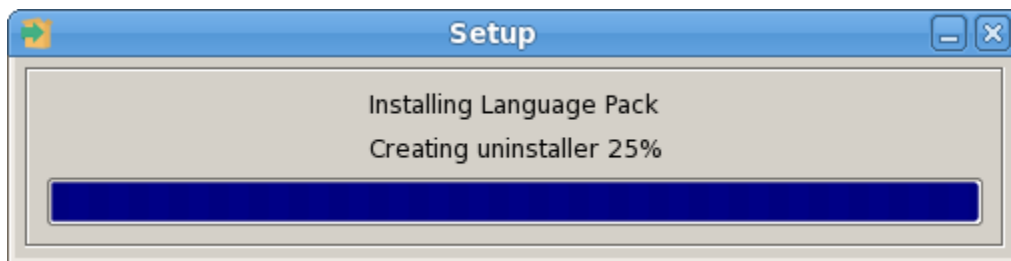


Figure 3.28 –The installer has detected missing dependencies.

After installing language pack, the installer will invoke the PEM-HTTPD setup wizard (see Figure 3.29).



Figure 3.29 - The PEM-HTTPD installation wizard.

Follow the onscreen instructions of the PEM-HTTPD Setup Wizard. When the wizard completes the installation, click **Finish** to open the Database Server Installation Details dialog.

The screenshot shows a Windows-style dialog box titled "Setup" with a subtitle "Database Server Installation Details". The main text reads: "Please specify the details of the database server on which Postgres Enterprise Manager's database component has been installed." Below this are four input fields: "Host" (192.168.2.135), "Port" (5445), "User" (enterprisedb), and "Password" (*****). At the bottom, there are three buttons: "< Back", "Next >", and "Cancel".

Figure 3.30 - The Database Server Installation Details dialog.

Use the fields on the Database Server Installation Details dialog (see Figure 3.30) to provide connection information for the Postgres installation that is hosting the PEM server installation:

- Enter the name or IP address of the PEM server host in the `Host` field.
- Enter the port number on which the Postgres server is listening in the `Port` field.
- Enter the name of a Postgres database superuser in the `User` field.
- Enter the password associated with the Postgres superuser in the `Password` field.

Click `Next` to continue. Before completing the PEM server installation, the installer will contact the database host. The `pg_hba.conf` file on the PEM database host must be configured to accept connections from the host of the httpd server and the firewall must allow a connection for the installation to continue. The PEM server installer will complete the PEM server installation, adding only those items that must reside on the host of the PEM-HTTPD server.

3.1.3 Installing the PEM Server on an Existing Postgres Server

You can use an existing Postgres server (version 9.4 or later) to host the PEM server and the `pem` database. Postgres installers and pre-requisite software extensions are freely available from the EnterpriseDB website at:

<http://www.enterprisedb.com/>

This section provides information about configuring an existing Postgres server for a PEM server installation. Please note that the steps that follow should be considered guidelines only; the actual steps required to configure your Postgres installation will vary depending on the configuration of your Postgres server.

The following versions of Postgres are pre-configured to contain the `sslutils` extension and a service script; no additional preparation is required to use the following Postgres versions as a PEM backend database server:

- PostgreSQL 9.4 or later (as bundled with the PEM Server installer)
- Advanced Server 9.4 or later

3.1.3.1 Preparing the Postgres Server

Before installing the PEM server on an existing Postgres server, you must:

- Ensure that the Postgres server contains an installation of the `sslutils` extension. For more information, see Section [3.1.3.1.1](#).
- Create a service script (on Linux) or register the server with the Windows service manager. For more information, see Section [3.1.3.1.2](#).

After preparing the server, you can use the PEM server installer to install PEM on the existing Postgres server.

3.1.3.1.1 Installing the `sslutils` Extension

The Postgres server on which the PEM server will reside must contain the `sslutils` extension. The `sslutils` package is freely available for download from the EnterpriseDB website at:

<https://www.enterprisedb.com/advanced-downloads>

When the web page opens, select the link for the `SRC-SSLUtils 1.2` package. When the download completes, extract the file, and copy it into the Postgres installation directory.

On Linux

If the Postgres server resides on a Linux system, you must use the `gcc` compiler to build `sslutils`.

1. Use `yum` to install `gcc`:

```
yum install gcc
```

2. Set the value of `PATH` so it can locate the `pg_config` program:

```
export PATH=$PATH:/opt/postgres_inst_dir/9.x/bin/
```

3. Then, use `yum` to install the `sslutil` dependencies:

```
yum install openssl-devel
```

4. Move into the `sslutils` folder, and enter:

```
make USE_PGXS=1
make USE_PGXS=1 install
```

5. Use `psql` to create the `sslutils` extension:

```
CREATE EXTENSION sslutils
```

On Windows

Remember: You are *not* required to manually add the `sslutils` extension when using the following Postgres installations:

- PostgreSQL 9.4 or later (as distributed with the PEM server installer)
- Advanced Server 9.4 or later

`sslutils` must be built with the same compiler that was used to compile the backend Postgres installation. If you are using a backend Postgres database that was installed on a Windows platform using a PostgreSQL one-click installer (from EnterpriseDB) or an Advanced Server installer, you must use Visual Studio to build `sslutils`.

While specific details of the installation process will vary by platform and compiler, the basic steps are the same. You must:

1. Copy the `sslutils` package to the Postgres installation directory.
2. Open the command line of the appropriate compiler, and navigate into the `sslutils` directory.
3. Use the following commands to build `sslutils`:

```
SET USE_PGXS=1
SET GETTEXT_PATH=path_to_gettext
SET OPENSSL_PATH=path_to_openssl
SET PGPATH=path_to_pg_installation_dir
SET ARCH=x86
REM Set ARCH x64 for 64 bit
msbuild sslutils.proj /p:Configuration=Release
```

Where:

`path_to_gettext` specifies the location of the `GETTEXT` library and header files.

`path_to_openssl` specifies the location of the `openssl` library and header files.

`path_to_pg_installation_dir` specifies the location of the Postgres installation.

4. Copy the compiled `sslutils` files to the appropriate directory for your installation. The `sslutils` directory will contain the following files:

```
sslutils--1.1.sql
sslutils--unpackaged--1.1.sql
sslutils--pemagent.sql.in
sslutils.dll
```

Copy the `.dll` libraries and `.sql` files into place:

```
COPY sslutils*.sql* "%PGPATH%\share\extension\"
COPY sslutils.dll "%PGPATH%\lib\"
```

3.1.3.1.2 Creating a Service Script or Registering the Service

A service script allows the PEM server to start, stop or restart the server if necessary when performing configuration management, certificate management, or other administrative tasks.

When you install a PostgreSQL or an Advanced Server database using an installer from EnterpriseDB (such as the PostgreSQL one-click installer), the installer will create a service script, or on Windows, register the service for you. If you have built the Postgres installation from source, you are required to manually create a service script.

While the PEM server installer checks for the presence of the service script, it does not check the integrity of the script itself; for PEM to function properly, you must ensure that the service script works as expected.

Writing a Linux Service Script

On Linux, the service script must reside in the `/etc/init.d` directory. The service script must be able to start, stop and restart the database server. Service scripts are platform-specific; you can find a sample service script for Linux in Section 7 Linux Service Script (Sample). For information about customizing a Postgres service, visit:

<https://www.postgresql.org/docs/current/static/server-start.html>

Registering a Service on Windows

If you are using Windows to host the PEM backend database, you must register the name of the Postgres server with the Windows service manager. If you are using a Postgres server that was created using an EnterpriseDB installer, the service will be registered automatically. If you are manually building the installation, you can use the `register` clause of the Postgres `pg_ctl` command to register the service. The syntax of the command is:

```
pg_ctl register [-N service_name] [-U user_name]
               [-P password] [-D data_directory]
```

Where:

service_name specifies the name of the Postgres cluster.

user_name specifies the name of an operating system user with sufficient privileges to access the Postgres installation directory and start the Postgres service.

password specifies the operating system password associated with the user.

data_directory specifies the location of the Postgres data directory.

For more information about using the `pg_ctl` command and the available command options, see the Postgres core documentation, available at:

<http://www.postgresql.org/docs/current/static/app-pg-ctl.html>

3.1.3.2 Invoking the PEM Server Installer

After preparing the existing Postgres server, invoke the PEM server installer. Assume superuser (or, on Windows, Administrative) privileges and navigate into the directory that contains the installer. Then, invoke the installer with the command:

```
./pem_server-7.x.x-x-platform.ext
```

The installer displays a `Welcome` dialog (shown in Figure 3.31).

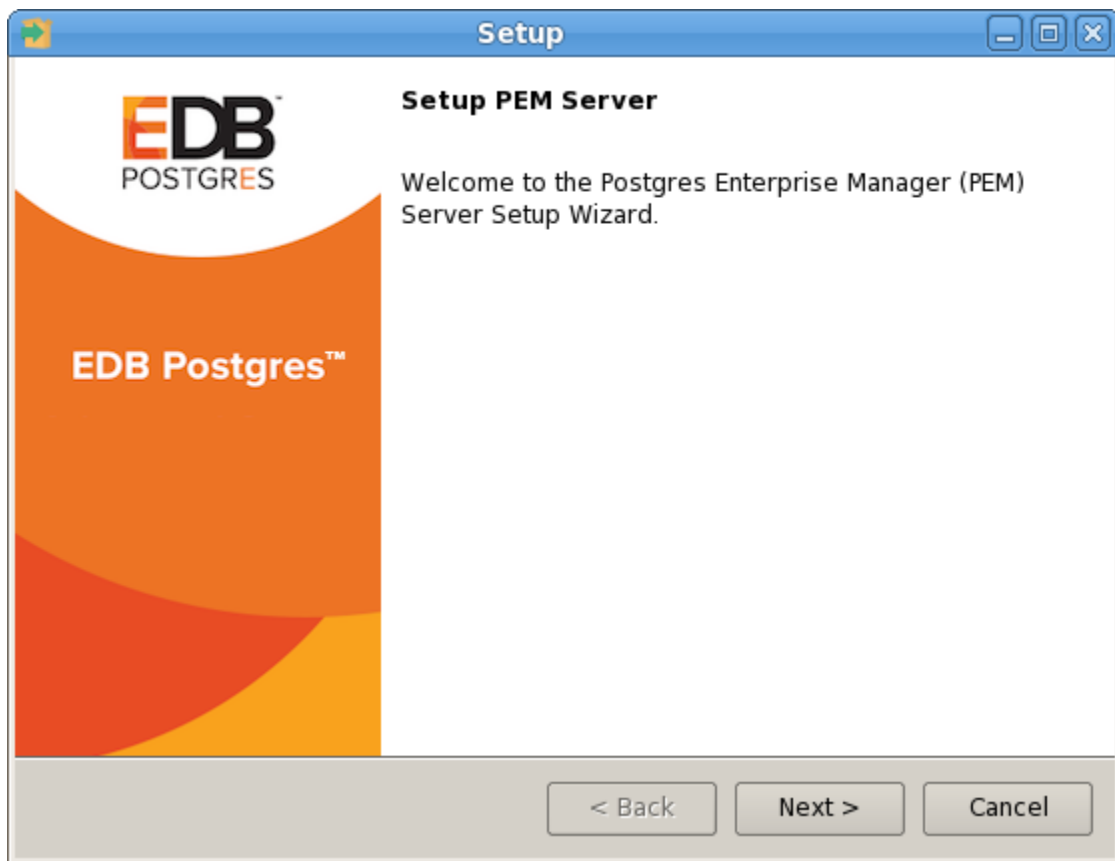


Figure 3.31 - The PEM Server Installer's Welcome dialog.

Click `Next` to continue to the `License Agreement` dialog.



Figure 3.32 - The License Agreement.

Carefully review the license agreement (see Figure 3.32) before highlighting the appropriate radio button and accepting the agreement; click `Next` to continue to the `Installation Directory` dialog.

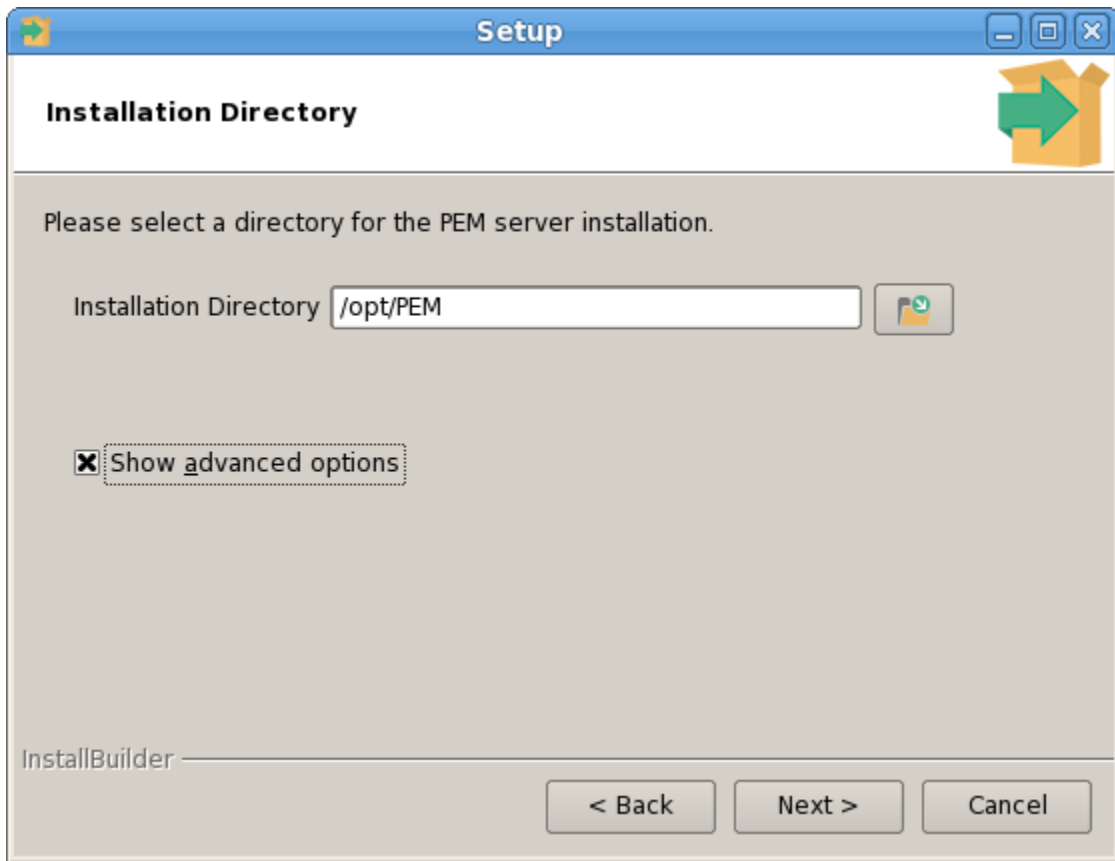


Figure 3.33 - The Installation Directory dialog.

Use the `Installation Directory` dialog (shown in Figure 3.33) to specify the location of the PEM server, and access the `Advanced options` dialog:

- Use the `Installation Directory` field to open a browser dialog and select the directory in which the PEM server will be installed.
- If you are installing the PEM server on an existing server, check the box next to `Show advanced options` to instruct the installer to include the `Advanced options` dialog in the installation process.

Click `Next` to continue.

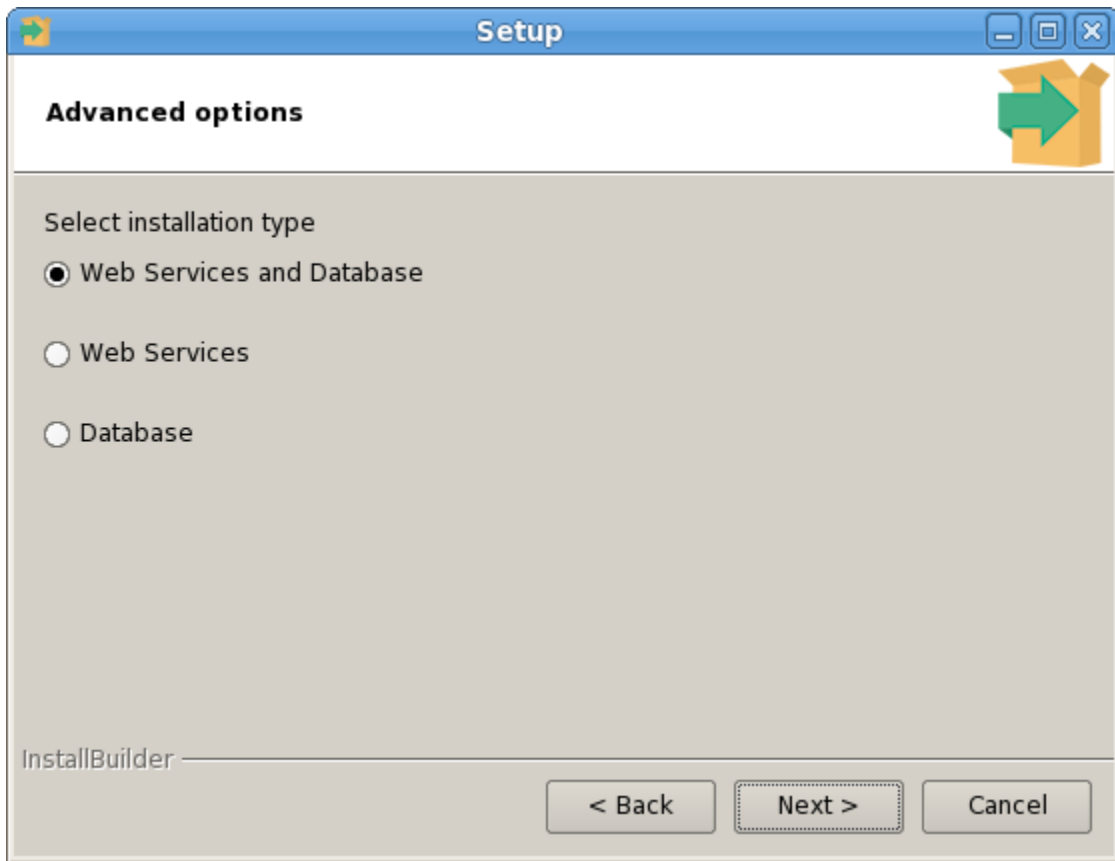


Figure 3.34 - The Advanced Options dialog.

Use the radio buttons on the `Advanced options` dialog (shown in Figure 3.34) to specify an installation type. Select:

- `Web Services and Database` if both the Postgres server and the PEM-HTTPD server will reside on the current host. This option is valid if you are using an existing Postgres server to host the PEM server, or using the PEM server installer to install the Postgres server on which the PEM server will reside.

If you select `Web Services and Database`, the PEM server installer will check the current host for a PEM-HTTPD installation, and upgrade or install PEM-HTTPD if necessary.

- `Web Services` if only the PEM-HTTPD server will reside on the current host. See [Section 3.1.2.2](#) for more information about invoking this option.
- `Database` if you are installing only the PEM server (and creating the `pem` backend database) on the current host. This option is valid if you are using an existing Postgres server to host the PEM server, or using the PEM server installer to install the PostgreSQL server on which PEM will reside.

After selecting an installation option, click **Next** to continue.

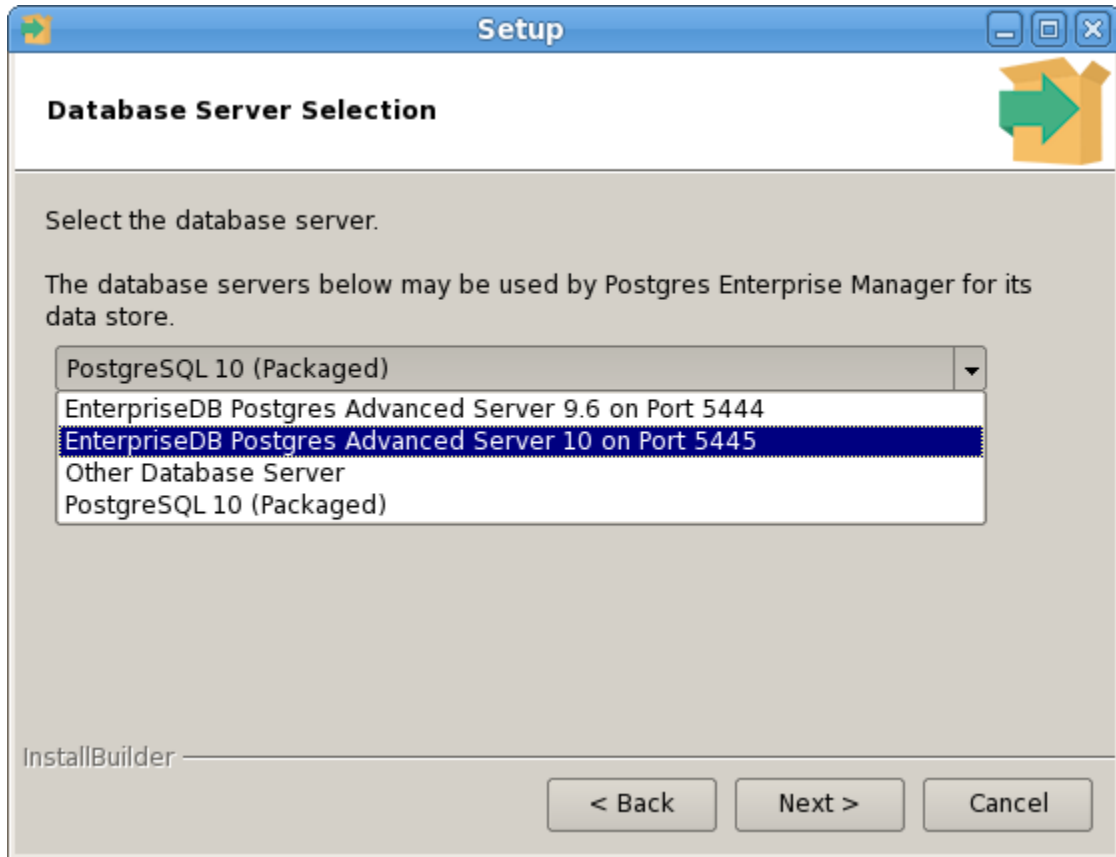


Figure 3.35 - The Database Server Selection dialog.

Use the drop-down listbox on the Database Server Selection dialog (Figure 3.35) to select a backend database for the PEM server:

- Select the name of a Postgres server on the current host that was installed using a Postgres one-click installer or Advanced Server installer.
- Select the `PostgreSQL 9.6 (Packaged)` option to instruct the installation wizard to install and use the PostgreSQL server that is packaged with the PEM server installer.
- Select `Other Database Server` to instruct the PEM server installer to use a Postgres database that was installed from a source other than an EnterpriseDB installer (i.e. from an rpm, or built from source).

Note that the selected database server must include an installation of the `sslutils contrib` module, and have a startup script (on Linux) or a registered service (on Windows). For information about *Preparing the Postgres Server*, please see [Section 3.1.3.1](#).

If you selected **Web Services** and **Database** on the **Advanced options dialog**, the installation wizard will check the current host for an existing PEM-HTTPD installation, and upgrade or install the service as needed.

If you selected **Database** on the **Advanced options dialog**, the **Database Server Installation Details dialog** opens (shown in Figure 3.36).

The screenshot shows a Windows-style dialog box titled "Setup" with a subtitle "Database Server Installation Details". The dialog contains the following text and fields:

Please specify the details of the database server on which Postgres Enterprise Manager's database component has been installed.

Host:

Port:

User:

Password:

At the bottom of the dialog, there are three buttons: "< Back", "Next >", and "Cancel". The "InstallBuilder" logo is visible in the bottom left corner.

Figure 3.36 - The Database Server Installation Details dialog.

Use the fields on the **Database Server Installation Details dialog** to describe the connection to the Postgres server that will host the PEM server:

- Enter the port number on which the Postgres server listens in the **Port** field.
- Enter the name of a database superuser in the **User** field.
- Enter the password associated with the superuser in the **Password** field.
- Enter the name of the service script for the Postgres server in the **Service Name** field.

Please note: For information about preparing a service script, please see [Section 3.1.3.1.2](#).

Click `Next` to continue.

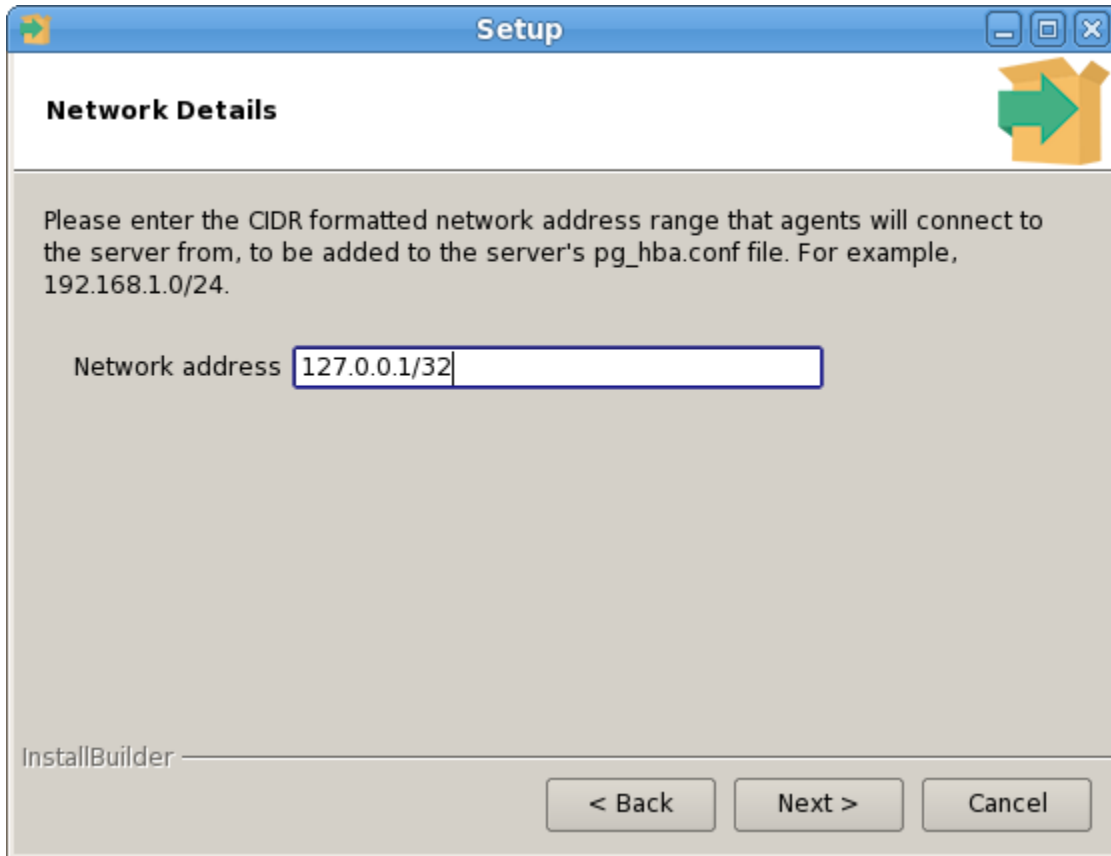


Figure 3.37 - Supply the network address from which the agent will connect.

Use the `Network Details` dialog (Figure 3.37) to specify the CIDR-style network address from which the PEM agents will connect to the server (the *client-side* address). The specified address will be added to the server's `pg_hba.conf` file.

You can specify additional network addresses by manually adding entries to the `pg_hba.conf` file on the PostgreSQL server if required, using the initial entry as a template.

When you've added the `Network address`, click `Next` to continue to the `Agent Details` dialog.

The PEM server installer will install a PEM agent to the host on which the server resides, to monitor the server and provide alert processing and garbage collection services. A certificate will also be installed in the location specified in the `Agent certificate path` field (see Figure 3.38).

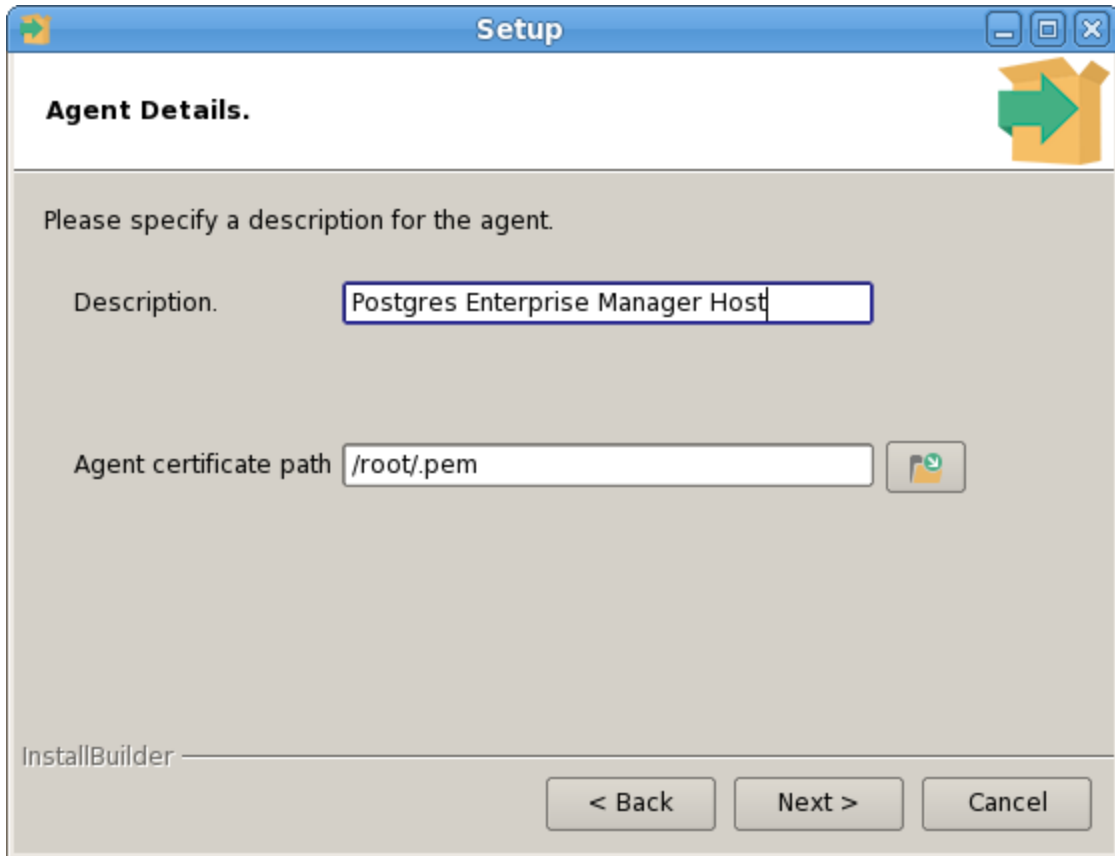


Figure 3.38 - Provide a Description for the agent that resides on the server.

You can enter an alternate description or an alternate agent certificate path for the PEM agent, or accept the defaults. Click **Next** to continue to the **Ready to Install** dialog.

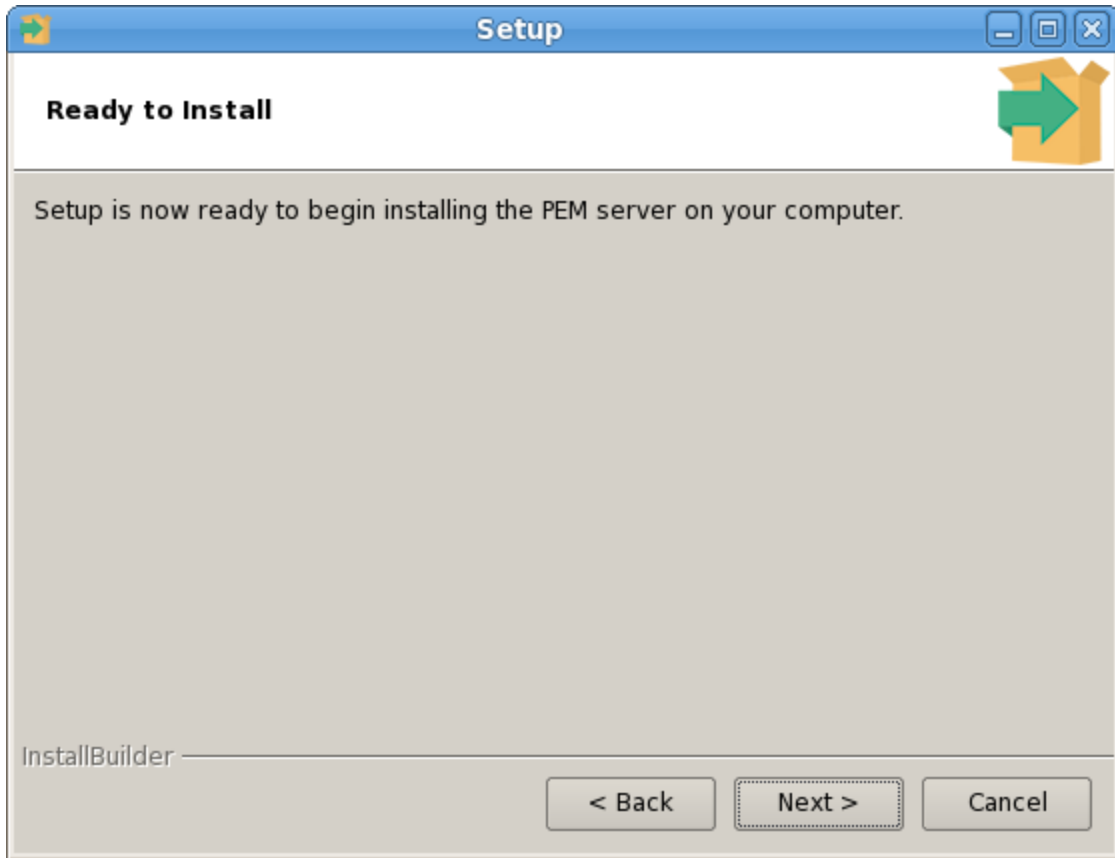


Figure 3.39 - The installation is ready to begin.

The wizard is now ready to install the PEM server (see Figure 3.39). Click `Back` to modify any of the options previously selected, or `Next` to continue with the installation.

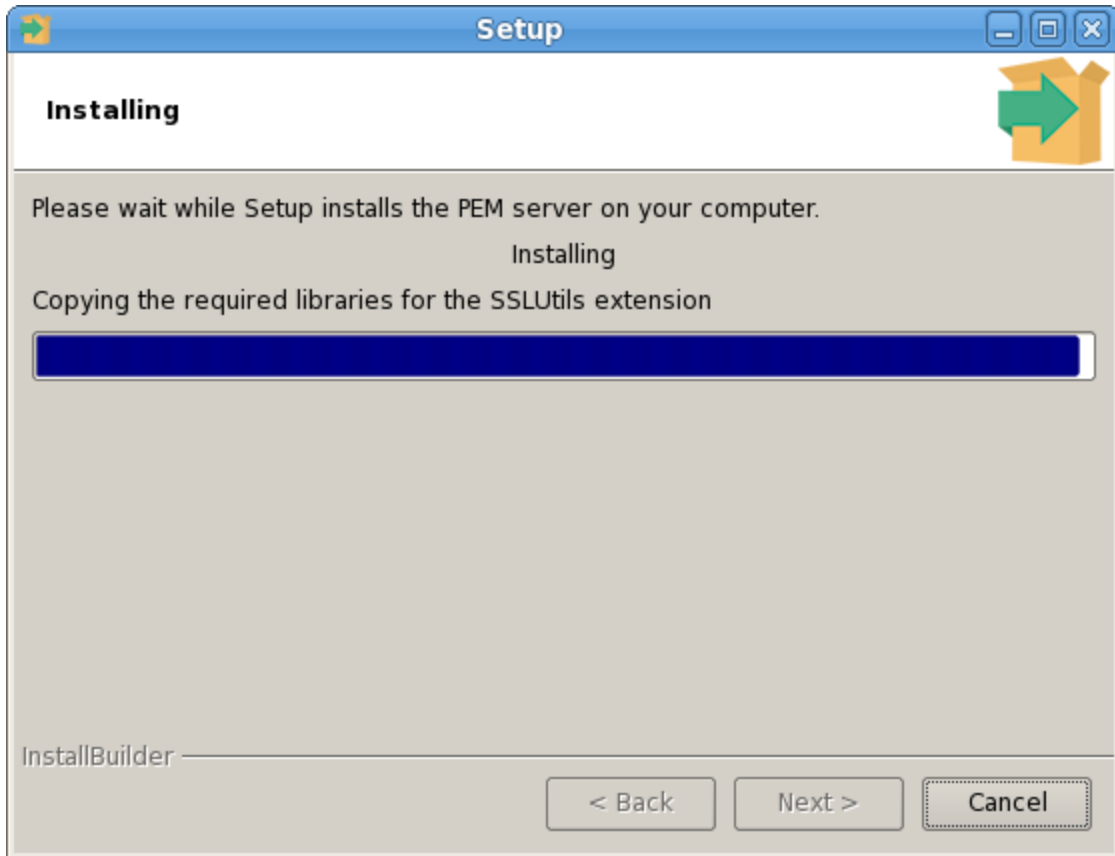


Figure 3.40 - The installation in progress.

During the installation process, the installer will copy files to the system, and set up the PEM server's backend database. A popup dialog opens (shown in Figure 3.41) confirming that the pem database has been created and configured.

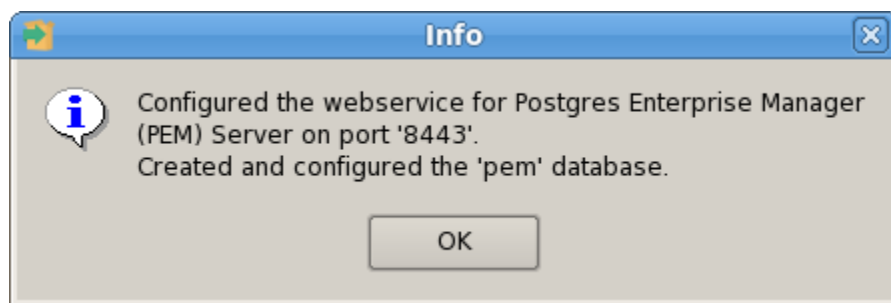


Figure 3.41 - Click OK to confirm that the pem database has been created.

Click OK to acknowledge that the pem database has been created, and continue to the Completed... dialog (shown in Figure 3.42).

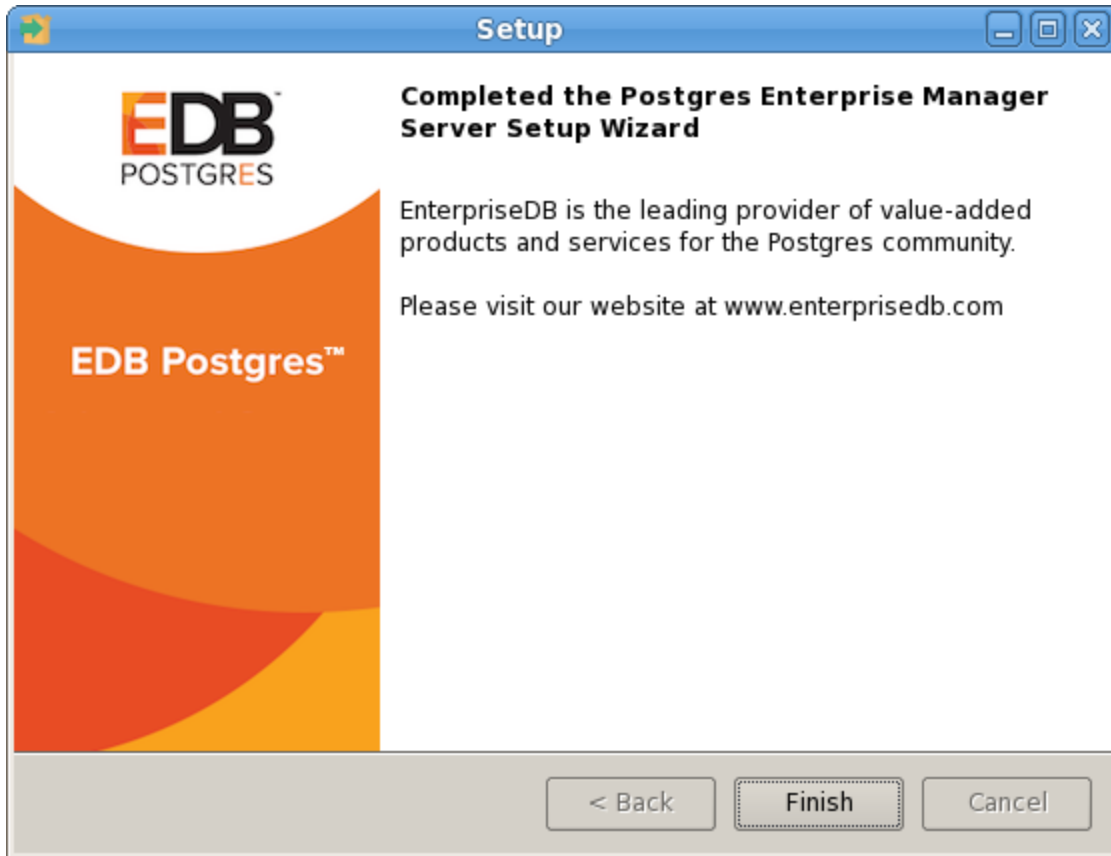


Figure 3.42 - The PEM server installation is complete.

If you are using a PEM-HTTPD service that resides on a separate host, you must:

- Modify the `pg_hba.conf` file on the Postgres server host to allow connections between the hosts.
- Invoke the PEM server installer on the host of the PEM-HTTPD server. See [Section 3.1.2.2](#) for more information about installing PEM-HTTPD.

3.2 Using an RPM Package to Install the PEM Server

You can use an RPM package on a RHEL or CentOS 6.x or 7.x host to install the PEM server and a PEM agent.

When using an RPM package to install the PEM server, you must first manually install a backing database and create the database cluster. The server's backing database must be installed via an RPM package or via BitRock. The database must be one of the following versions:

- EDB Postgres Advanced Server version 9.4 and above
- PostgreSQL version 9.4 and above

For detailed information about installing an Advanced Server or PostgreSQL database, please see the product documentation at:

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

The `pg_hba.conf` file on the backing database must be configured to use trust authentication for connections. For information about modifying the `pg_hba.conf` file, visit:

<https://www.postgresql.org/docs/current/static/auth-pg-hba-conf.html>

You must also install the `hstore contrib` module; for more information, visit:

<https://www.postgresql.org/docs/current/static/hstore.html>

Please note that the installer will not upgrade an existing agent; you must manually remove an existing agent.

PEM's Package Deployment and Streaming Replication Deployment/Configuration features are designed to work with the EnterpriseDB installer packages, not platform native packaging packages. While those features will still work with a natively packaged version of PEM (such as an RPM or DEB), the features can only deploy installer packages and not RPM or DEB packages. You should instead use vendor supplied management tools such Yum or Aptitude to deploy native packages, in conjunction with configuration management tools such as Chef, Puppet, SaltStack or Ansible if desired.

You can also use EDB Ark to deploy and manage Postgres clusters using streaming replication on cloud platforms using native packages.

Before installing the PEM server, you must create the repository configuration file (`edb.repo`). The repository configuration file contains connection and authentication information for the EnterpriseDB repository. To prepare your system to perform an RPM

installation, assume superuser privileges and use `yum` to create the repository configuration file:

```
yum install -y http://yum.enterprisedb.com/edbrepos/edb-repo-latest.noarch.rpm
```

You must also install the `epel-release` package. On a CentOS host, use the command:

```
yum install epel-release
```

If you are using a RHEL host, use the following command:

```
yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

Please note: If you are a Red Hat Network user you must also enable the `rhel-x-server-optional-rpms` repository to use EPEL packages, where `x` specifies the version of RHEL on the host. You can make the repository accessible by enabling the RHEL optional subchannel for RHN-Classic. If you have a certificate-based subscription, please see the Red Hat Subscription Management Guide. You must also enable the `rhel-x-server-extras-rpms` repository, where `x` specifies the version of RHEL on the host.

After creating the repository configuration file, use your choice of editor to enable the repository entries from which you will install packages. The repository configuration file is named `edb.repo`, and resides in `/etc/yum.repos.d`. To enable a repository, change the value of the `enabled` parameter to `1` and replace the username and password placeholders in the `baseurl` specification with your user name and the repository password.

If you need credentials for the EnterpriseDB repository, please contact EnterpriseDB.

Enable the following repositories:

```
[enterprisedb-tools]
[enterprisedb-dependencies]
```

Note that if you are using a RHEL or CentOS 6.x host, you need to stop the default `httpd` server and also run commands to install `rh-python36` before installing the PEM server. As part of PEM server installation, `httpd24` server gets installed if you are using a RHEL or CentOS 6.x host.

Stop the default `httpd` server using the command:

```
service httpd stop
```

Run following commands to install `rh-python36`:

```
yum install centos-release-scl  
  
yum install rh-python36
```

Before installing the PEM server, you must disable `SELinux`; use the command:

```
setenforce 0
```

If you are using a firewall, you must allow access to port 8443; use the commands:

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

Then, you can use `yum` to install the PEM server:

```
yum install edb-pem-server
```

The installer will install a web services component; use the following command to instruct the web server to start when the system boots:

```
systemctl enable httpd
```

Before configuring the PEM server, ensure that the `sslutils` extension is installed for your backing database. For an Advanced Server backing database, enable the repository that corresponds to your server version, and use the command:

```
yum install edb-asX-server-sslutils
```

Where, `X` is the server version. If you are using a PostgreSQL backing database, ensure you have access to the PostgreSQL community repository, and use the command:

```
yum install sslutils_X postgresqlX-contrib
```

After installing the backing database and the PEM server, you must configure the installation.

Configuring the PEM Server

If you are using a RHEL or CentOS 6.x host, you need to set the default python path before configuring the PEM server; use the command:

```
scl enable rh-python36 bash
```

This command sets the default python path as:

```
/opt/rh/rh-python36/root/usr/bin/python
```

The PEM server installer includes a script (`configure-pem-server.sh`) to help automate the configuration process for RPM installations. The script is installed in the `/usr/edb/pem/bin` directory. To invoke the script, use the command:

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

When invoking the script, you can include command line options to specify configuration properties; the script will prompt you for values that you omit on the command line. The accepted options are:

Option	Description
<code>-acp</code>	Defines PEM Agent certificate path. The default is <code>/root/.pem</code> .
<code>-ci</code>	CIDR formatted network address range that agents will connect to the server from, to be added to the server's <code>pg_hba.conf</code> file. For example, <code>192.168.1.0/24</code> . The default is <code>0.0.0.0/0</code> .
<code>-dbi</code>	The directory for the database server installation. For example, <code>/usr/edb/as10</code> for Advanced Server or <code>/usr/pgsql-10</code> for PostgreSQL.
<code>-ds</code>	The unit file name of the PEM database server. For Advanced Server, the default file name is <code>edb-as-10</code> ; for PostgreSQL, it is <code>postgresql-10</code> .
<code>-ho</code>	The host address of the PEM database server.
<code>-p</code>	The port number of the PEM database server.
<code>-ps</code>	The service name of the pemagent; the default value is <code>pemagent</code> .
<code>-sp</code>	The superuser password of the PEM database server. This value is required.
<code>-su</code>	The superuser name of the PEM database server.
<code>-t</code>	The installation type: Specify <code>1</code> if the configuration is for web services and backing database, <code>2</code> if you are configuring web services, or <code>3</code> if you are configuring the backing database. If you specify <code>3</code> , please note that the database must reside on the local host.

If you do not provide configuration properties on the command line, you will be prompted for values by the script. When you invoke the script, choose from:

1. Web Services and Database

Select this option if the web server and database both reside on the same host as the PEM server.

2. Web Services

Select this option if the web server resides on a different host than the PEM server.

3. Database

Select this option to configure the PEM backing database for use by the PEM server. Please note that the specified database must reside on the local host.

Please note that if the web server and the backing database reside on separate hosts, configure the database server first (option 3), and then web services (option 2). The script will exit if the backing database is not configured before web services.

After selecting a configuration option, the script will proceed to prompt you for configuration properties. When the script completes, it will create the objects required by the PEM server, or perform the configuration steps required.

To view script-related help, use the command:

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

If you are using a RHEL or CentOS 6.x host, restart the `httpd24` server after configuring the PEM server; use the command:

```
service httpd24-httpd restart
```

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

```
https://ip_address_of_PEM_server:8443
```

3.2.1 Installing on a SLES Host

For detailed information about installing Advanced Server and supporting components on a SLES host, please consult the EDB Postgres Advanced Server Installation Guide, available at:

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

SLES packages are available from:

<https://zypp.enterprisedb.com>

Before installing PEM, you must install prerequisite packages. If you are using a SLES SP2 host, use the following commands:

```
SUSEConnect -p sle-module-legacy/12/x86_64
SUSEConnect -p sle-sdk/12/x86_64
zypper addrepo
https://download.opensuse.org/repositories/Apache:Modules/SLE_12_
SP2/Apache:Modules.repo
zypper addrepo
http://download.opensuse.org/repositories/Cloud:/OpenStack:/Newto
n:/cisco-apic:/2.3.1/SLE_12_SP2/ pem_opensuse_boost
zypper refresh
```

If you are using a SP3 host, use the following commands:

```
SUSEConnect -p sle-module-legacy/12/x86_64
SUSEConnect -p sle-sdk/12/x86_64
zypper addrepo
https://download.opensuse.org/repositories/Apache:Modules/SLE_12_
SP3/Apache:Modules.repo
zypper addrepo
http://download.opensuse.org/repositories/Cloud:/OpenStack:/Newto
n:/cisco-apic:/2.3.1/SLE_12_SP3/ pem_opensuse_boost
zypper refresh
```

3.2.2 Installing on a Debian or Ubuntu Host

To install PEM on a Debian or Ubuntu host, you must have credentials that allow access to the [EnterpriseDB repository](#). To request credentials for the repository, visit the EnterpriseDB Advanced Downloads page at:

<https://www.enterprisedb.com/advanced-downloads>

The following steps will walk you through using the EnterpriseDB apt repository to install a Debian package. When using the commands, replace the username and password with the credentials provided by EnterpriseDB.

1. Go to <https://apt.enterprisedb.com/> and log in as root:

```
sudo su -
```

2. Configure the EnterpriseDB repository:

```
sh -c 'echo "deb
https://username:password@apt.enterprisedb.com/$(lsb_release -
cs)-edb/ $(lsb_release -cs) main" > /etc/apt/sources.list.d/edb-
$(lsb_release -cs).list'
```

3. Add support to your system for secure APT repositories:

```
apt-get install apt-transport-https
```

4. Add the EBD signing key:

```
wget -q -O - https://username:password@apt.enterprisedb.com/edb-
deb.gpg.key | apt-key add -
```

5. Update the repository metadata:

```
apt-get update
```

6. Use the following command to install the Debian package:

- For PEM Server and agent: `apt-get install edb-pem-server`
- For PEM agent: `apt-get install edb-pem-agent`

3.3 Installing a PEM Agent

The PEM agent executes tasks scheduled via the PEM server, and reports statistics from the agent's operating system and one or more monitored Postgres servers.

3.3.1 Using the Graphical Installer to Install an Agent

To invoke the PEM agent installer, assume superuser (or Administrative) privileges and navigate into the directory that contains the installer. Then, invoke the installer with the command:

```
pem_agent-7.x.x-x-platform.ext
```

The Setup... page opens, welcoming you to the PEM Agent installer (as shown in Figure 3.43).



Figure 3.43 - The PEM Agent Installer's Welcome dialog.

Click **Next** to continue to the **License Agreement**.



Figure 3.44 - The PEM License Agreement.

Carefully review the license agreement (see Figure 3.44) before highlighting the appropriate radio button and accepting the agreement; click `Next` to continue to the `Installation Directory` dialog.

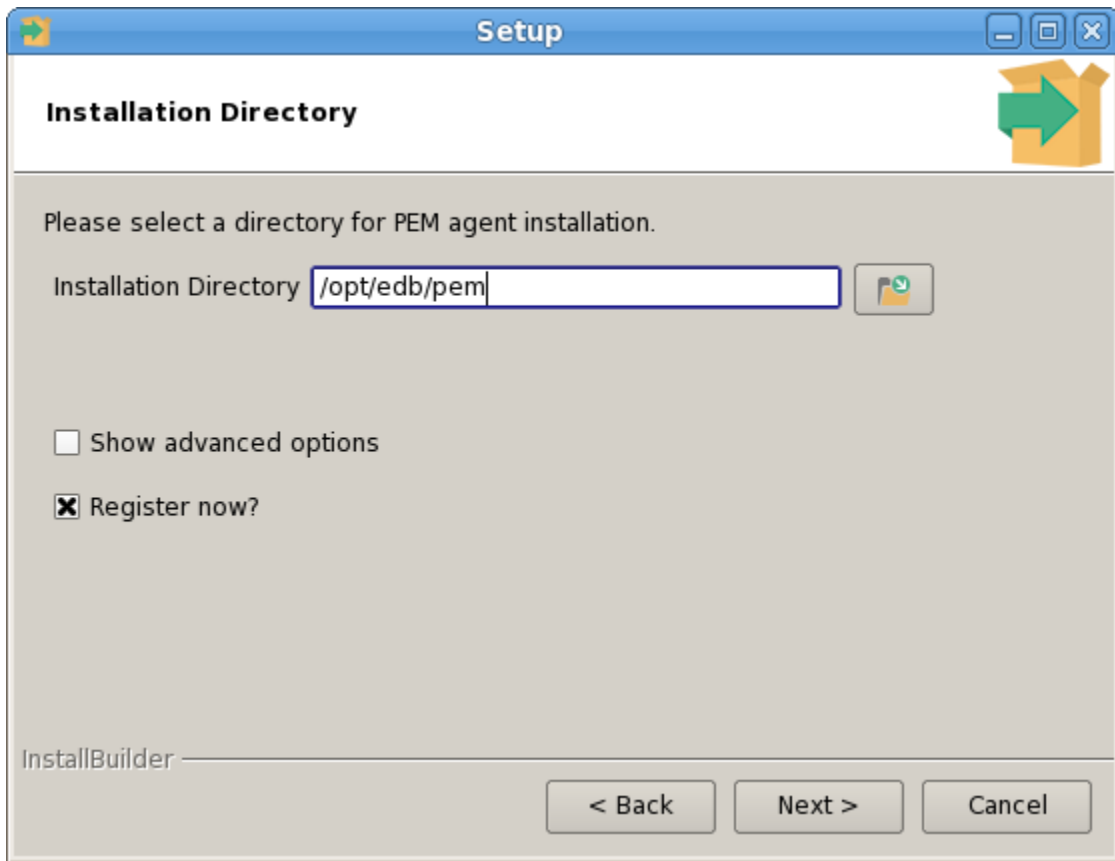


Figure 3.45 - Specify an Installation Directory

By default, the PEM agent is installed in the `/home/opt/PEM` directory. You can accept the default installation directory, or modify the contents of the `Installation Directory` field (see Figure 3.45), specifying an alternate installation directory for the PEM agent.

By default, the PEM agent installer places a certificate in `~root/.pem` (on Linux) or in the Administrator's `%APPDATA%\pem` directory (on Windows). Check the `Show advanced options` box to indicate that you would like the PEM agent installer to include a dialog that allows you to specify an alternate path for the certificate file.

Check the box next to `Register now?` to instruct the installer to register the newly installed PEM agent with the PEM server.

Click `Next` to continue to the `PEM Server Installation Details` dialog.

Setup

PEM server installation details

Please verify the PEM server installation details

Host

User Name

Password

Port

InstallBuilder

< Back Next > Cancel

Figure 3.46 - Provide PEM server installation details

Enter the connection details for the PEM server on the `PEM server installation details` dialog (see Figure 3.46):

- Specify the name or IP address of the system on which the PEM database server resides in the `Host` field. Please note: If the PEM-HTTPD web server and PEM database are hosted on different systems, you must specify *the host of the PEM database*.
- Specify the name of the database superuser in the `User Name` field.
- Specify the password associated with the database superuser in the `Password` field.
- Specify the port that PostgreSQL is monitoring in the `Port` field.

Click `Next` to continue. The installer will attempt to connect to the server to verify that the details are correct.

Please Note: The PEM server must allow connections from the PEM agent installer. If you encounter a connection error, confirm the connection properties specified on the `PEM Server Installation Details` dialog are correct, and confirm that the `pg_hba.conf` file (on the PEM server) will allow a connection to the server described in the error message.

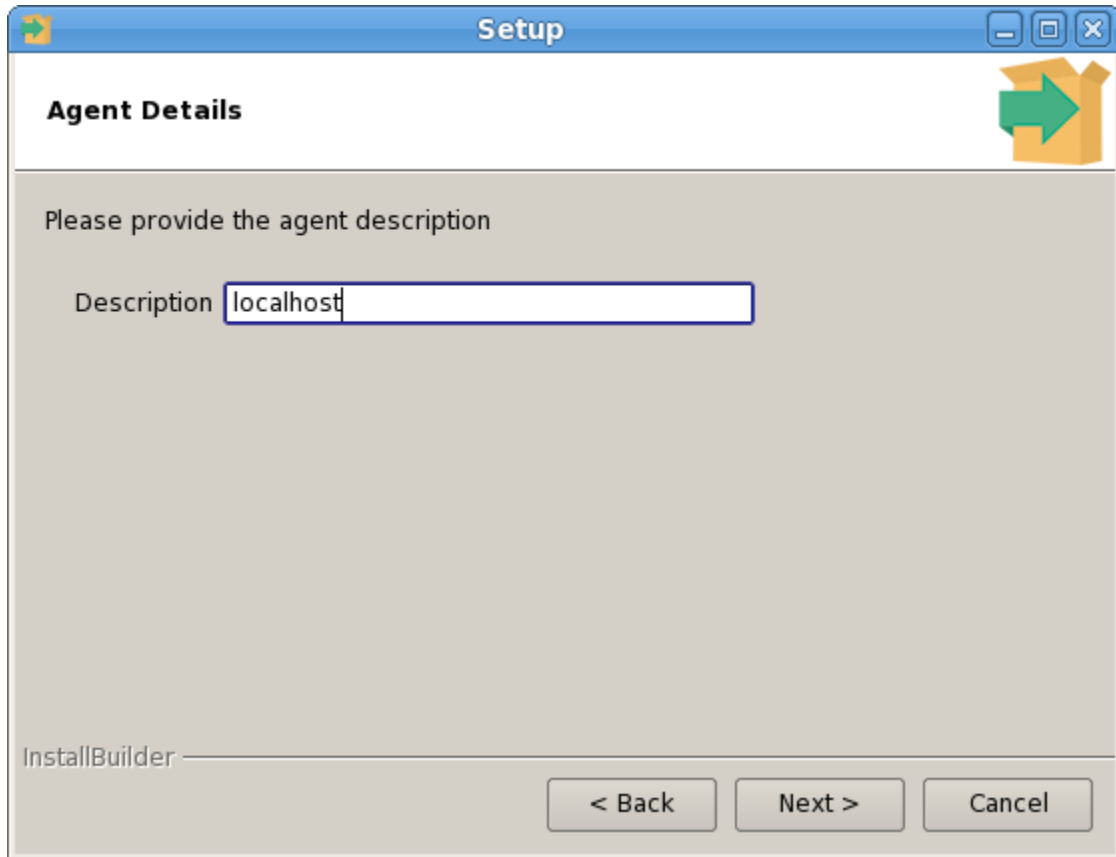


Figure 3.47 - Specify a descriptive name for the PEM agent

The tree control displayed in the `Browser` panel of the PEM web interface displays the value entered in the `Description` field (shown in Figure 3.47) to identify the PEM agent. Specify a descriptive name for the agent, such as the hostname of the machine the agent is installed on, or a name that reflects the host's functionality.

Provide a descriptive name, or accept the default provided by the PEM agent host, and click `Next` to continue.

If you checked the `Show advanced options` checkbox shown in Figure 3.24, the `Advanced options` dialog opens (shown in Figure 3.48).

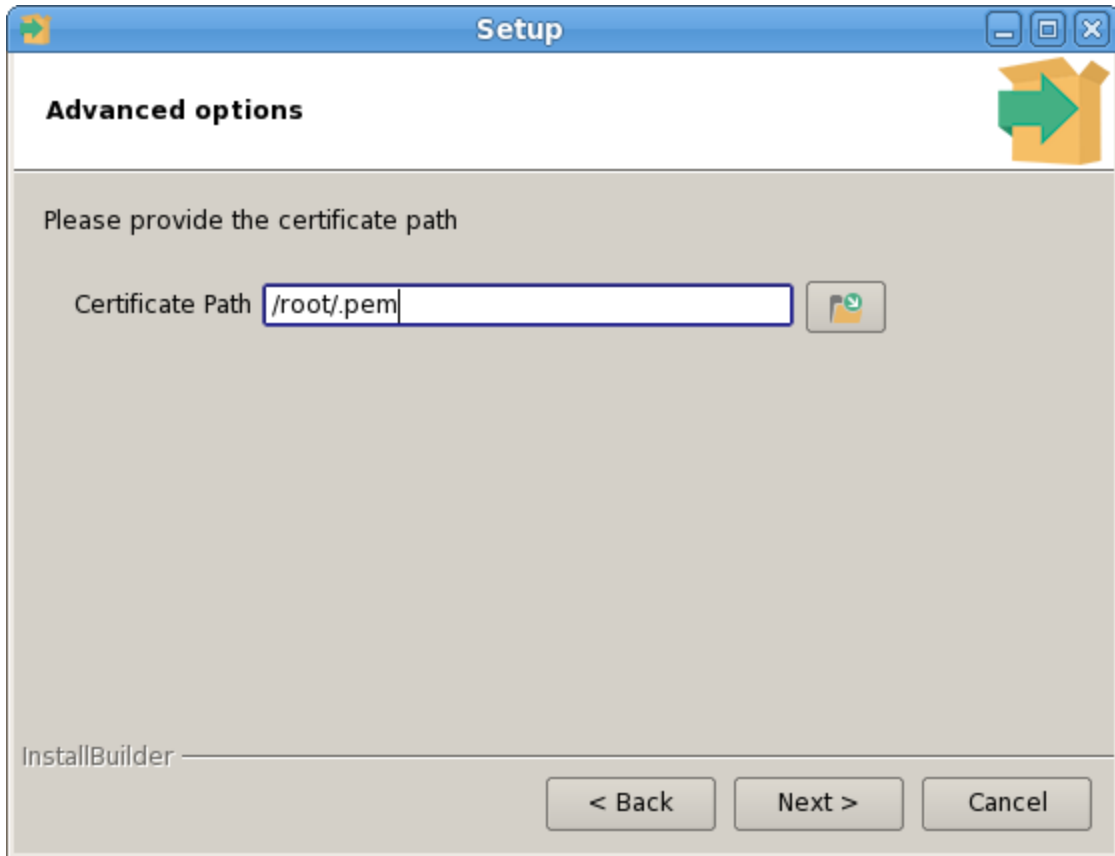


Figure 3.48 – Provide the certificate path

By default, the PEM agent installer places the certificate in the `/root/.pem` directory. Specify an alternate path for the certificate or accept the default and click `Next`.

The wizard is now ready to install the PEM agent; click `Back` to amend the installation directory, or `Next` to continue.

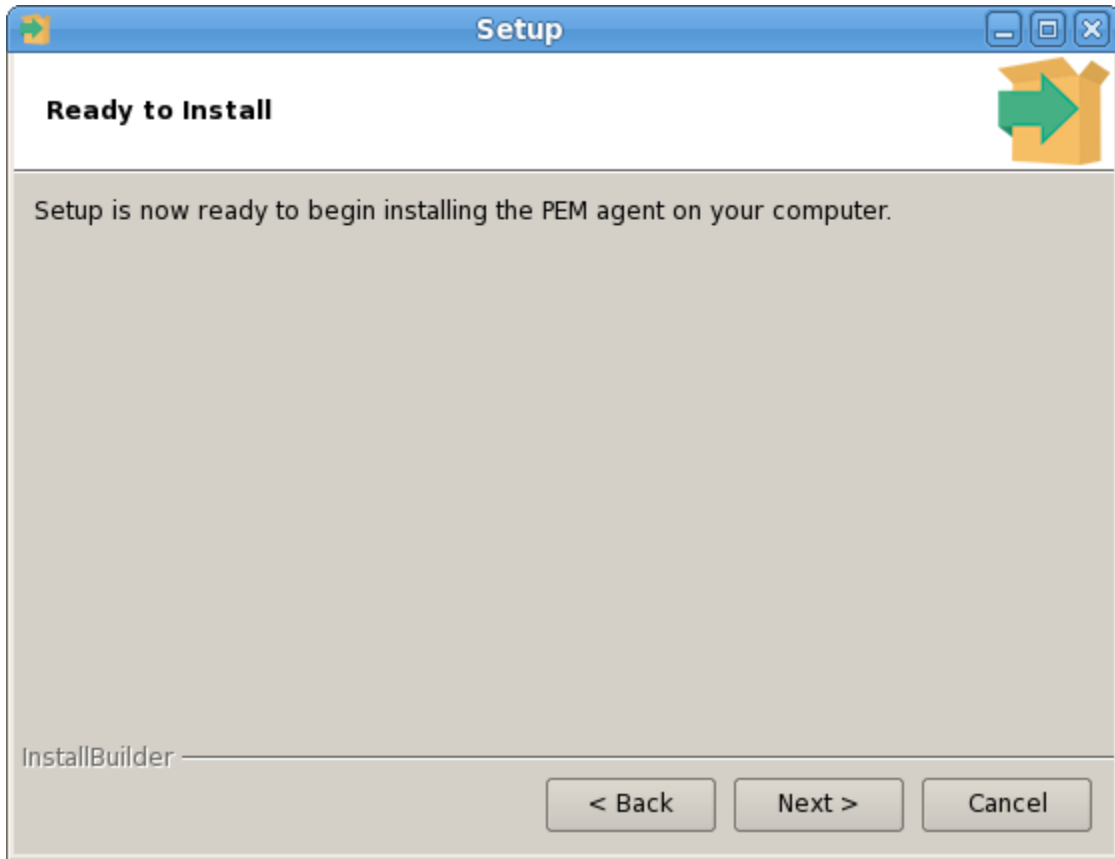


Figure 3.49 - The PEM Agent installation is ready to begin

Click **Next** on the **Ready to Install** dialog (shown in Figure 3.49) to instruct the installer to copy files to the system and register the agent on the PEM server.

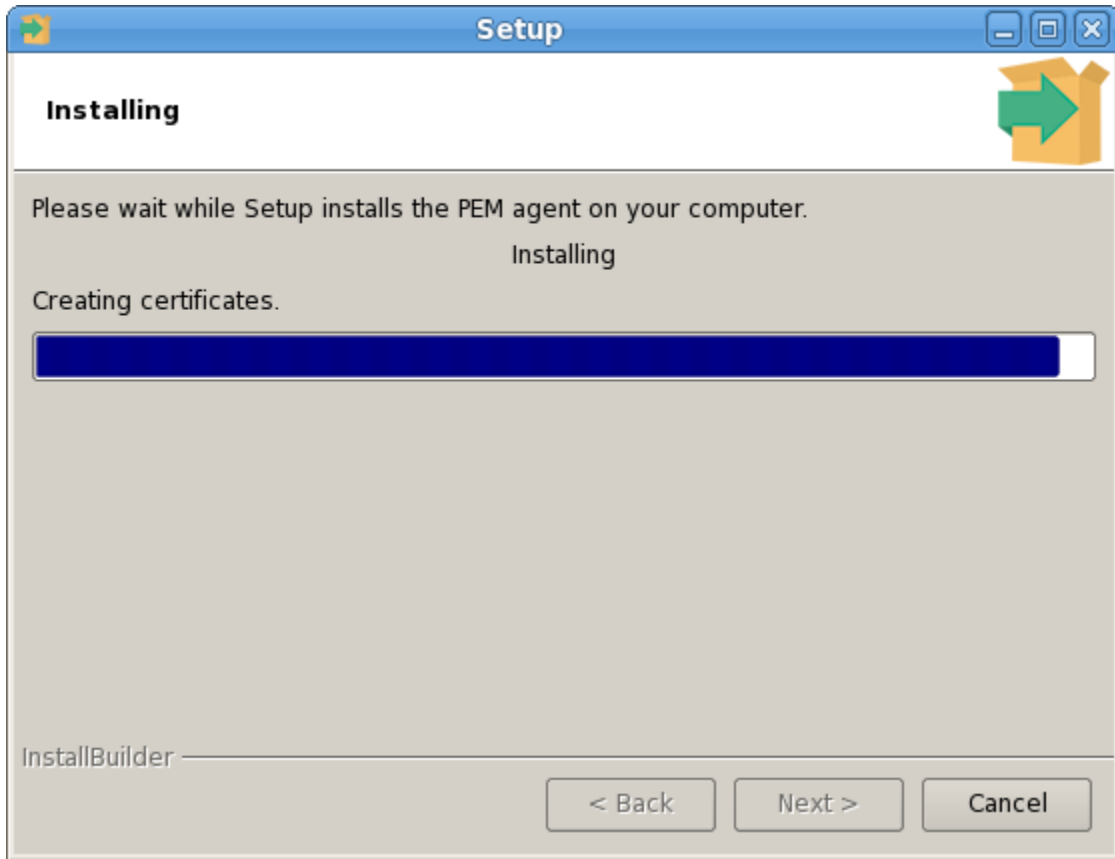


Figure 3.50 - Progress bars mark the installation's progress

The PEM agent installer displays progress bars (shown in Figure 3.50) to mark the PEM agent's installation progress.

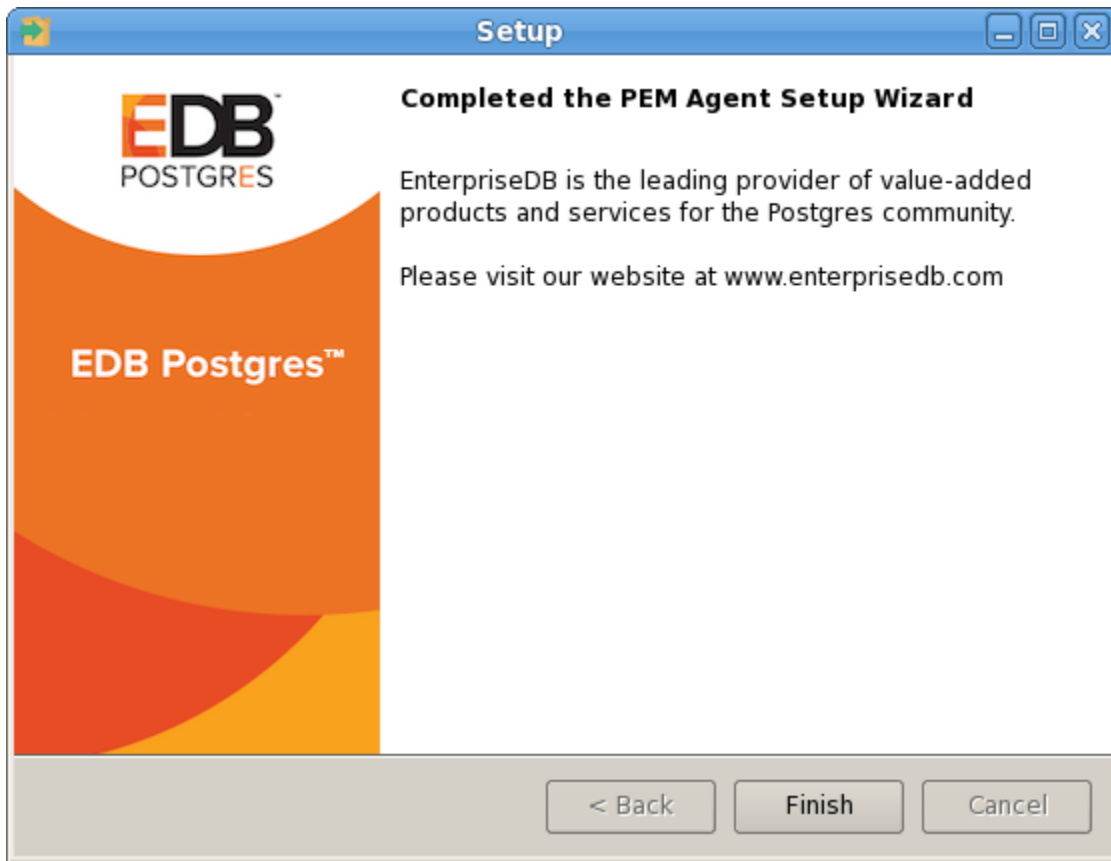


Figure 3.51 - The PEM Agent installation is complete

When the installation has completed (see Figure 3.51), the PEM agent will be running and reporting operating system and host data to the PEM server. To start monitoring Postgres instances on the host of the PEM agent, they must now be added to PEM's enterprise directory and bound to the agent.

3.3.2 Installing an Agent with an RPM Package

On a Linux system, you can use the `yum` package manager to install a PEM agent. Please note that before using a package manager to install the PEM agent, the host must contain the following packages:

```
epel-release
wxBase
```

After providing the pre-requisite packages, you can install the PEM agent:

1. Download the `edb-repo` installation package from:

<http://yum.enterprisedb.com/>

The `edb-repo` package creates the repository configuration file (named `edb.repo`). The `edb.repo` file defines multiple repositories hosted at EnterpriseDB.com.

2. Assume superuser privileges and use the following command to install the `edb-repo` package, and create the repository configuration file:

```
rpm -Uvh edb-repo-9.x-x.noarch.rpm
```

Where `x` specifies the version of the file.

Then, use your choice of editor to modify the configuration file, enabling the `enterprisedb-tools` and `enterprisedb-dependencies` repositories. The configuration file is named `edb.repo`; it resides in `/etc/yum.repos.d`.

To enable a repository, change the value of the `enabled` parameter to 1 and replace the `user_name` and `password` placeholders in the `baseurl` specification with your repository credentials. Contact [EnterpriseDB](#) for repository credentials.

For example:

```
[enterprisedb-tools]
name=EnterpriseDB Tools $releasever - $basearch
baseurl=http://user_name:password@yum.enterprisedb.com/tools/r
edhat/rhel-$releasever-$basearch
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/ENTERPRISEDB-GPG-KEY

[enterprisedb-dependencies]
name=EnterpriseDB Dependencies $releasever - $basearch
```

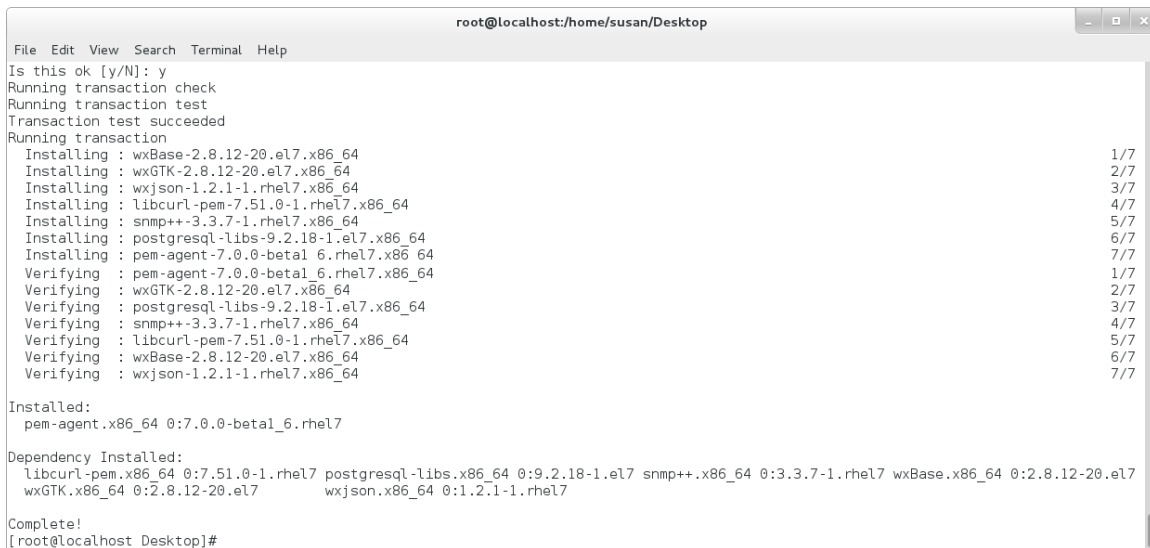
```
baseurl=http://user_name:password@yum.enterprisedb.com/dependencies/redhat/rhel-$releasever-$basearch
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/ENTERPRISEDB-GPG-KEY
```

```
[edbas96]
name=EnterpriseDB Advanced Server 9.6 $releasever - $basearch
baseurl=http://user_name:password@yum.enterprisedb.com/9.6/redhat/rhel-$releasever-$basearch
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/ENTERPRISEDB-GPG-KEY
```

3. After modifying the content of the repository configuration file, you can use yum to install the PEM agent:

```
yum install edb-pem-agent
```

When the installation is complete, yum will display a list of the installed packages and dependencies (see Figure 3.52).



```
root@localhost:/home/susan/Desktop
File Edit View Search Terminal Help
Is this ok [y/N]: y
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : wxBase-2.8.12-20.el7.x86_64 1/7
Installing : wxGTK-2.8.12-20.el7.x86_64 2/7
Installing : wxjson-1.2.1-1.rhel7.x86_64 3/7
Installing : libcurl-pem-7.51.0-1.rhel7.x86_64 4/7
Installing : snmp++-3.3.7-1.rhel7.x86_64 5/7
Installing : postgresql-libs-9.2.18-1.el7.x86_64 6/7
Installing : pem-agent-7.0.0-beta1_6.rhel7.x86_64 7/7
Verifying : pem-agent-7.0.0-beta1_6.rhel7.x86_64 1/7
Verifying : wxGTK-2.8.12-20.el7.x86_64 2/7
Verifying : postgresql-libs-9.2.18-1.el7.x86_64 3/7
Verifying : snmp++-3.3.7-1.rhel7.x86_64 4/7
Verifying : libcurl-pem-7.51.0-1.rhel7.x86_64 5/7
Verifying : wxBase-2.8.12-20.el7.x86_64 6/7
Verifying : wxjson-1.2.1-1.rhel7.x86_64 7/7

Installed:
pem-agent.x86_64 0:7.0.0-beta1_6.rhel7

Dependency Installed:
libcurl-pem.x86_64 0:7.51.0-1.rhel7 postgresql-libs.x86_64 0:9.2.18-1.el7 snmp++.x86_64 0:3.3.7-1.rhel7 wxBase.x86_64 0:2.8.12-20.el7
wxGTK.x86_64 0:2.8.12-20.el7 wxjson.x86_64 0:1.2.1-1.rhel7

Complete!
[root@localhost Desktop]#
```

Figure 3.52 – Using an RPM package to install the PEM agent.

When you install an RPM package that is signed by a source that is not recognized by your system, yum may ask for your permission to import the key to your local server. If prompted, and you are satisfied that the packages come from a trustworthy source, enter a y, and press Return to continue.

During the installation, yum may encounter a dependency that it cannot resolve. If it does, it will provide a list of the required dependencies that you must manually resolve.

3.3.3 Installing on a SLES Host

For detailed information about installing Advanced Server and supporting components on a SLES host, please consult the EDB Postgres Advanced Server Installation Guide, available at:

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

SLES packages are available from:

<https://zypp.enterprisedb.com>

Before installing a PEM agent, you must install prerequisite packages. If you are using a SLES SP2 host, use the following commands:

```
SUSEConnect -p sle-module-legacy/12/x86_64
SUSEConnect -p sle-sdk/12/x86_64
zypper addrepo
https://download.opensuse.org/repositories/Adobe:Modules/SLE_12_SP2/Adobe:Modules.repo
zypper addrepo
http://download.opensuse.org/repositories/Cloud:/OpenStack:/Newton:/cisco-apic:/2.3.1/SLE_12_SP2/ pem_opensuse_boost
zypper refresh
zypper install edb-pem-agent
```

If you are using a SLES 12 SP3 host, use the following commands in the given sequence to install the agent:

```
SUSEConnect -p sle-module-legacy/12/x86_64
SUSEConnect -p sle-sdk/12/x86_64
zypper addrepo
https://download.opensuse.org/repositories/Adobe:Modules/SLE_12_SP3/Adobe:Modules.repo
zypper addrepo
http://download.opensuse.org/repositories/Cloud:/OpenStack:/Newton:/cisco-apic:/2.3.1/SLE_12_SP3/ pem_opensuse_boost
zypper refresh
zypper install edb-pem-agent
```

3.3.4 Registering an Agent

Each PEM agent must be *registered* with the PEM server. The registration process provides the PEM server with the information it needs to communicate with the agent. The PEM agent graphical installer supports agent self-registration, but you can use the `pemworker` utility to register the agent if you skip PEM agent registration during a graphical installation or use an RPM package to install a PEM agent.

The RPM installer places the PEM agent in the `/usr/pem/agent/bin` directory. To register an agent, include the `--register-agent` keywords along with registration details when invoking the `pemworker` utility:

```
pemworker --register-agent
```

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

Option	Description
<code>--pem-server</code>	Specifies the IP address of the PEM server. This parameter is required.
<code>--pem-user</code>	Specifies the name of the PEM user. This parameter is required.
<code>--pem-port</code>	Specifies the port that PEM monitors for connections. The default value is 5432.
<code>--cert-path</code>	Specifies the complete path to the directory in which certificates will be created. If you do not provide a path, certificates will be created in: On Linux, <code>~/pem</code> On Windows, <code>%APPDATA%/pem</code>
<code>--display-name</code>	Specifies a user-friendly name that will be displayed in the PEM Browser tree control. The default is the system hostname.
<code>--group</code>	The name of the group in which the agent will be displayed.
<code>--team</code>	The name of the group role that may access the PEM Agent.
<code>--owner</code>	The name of the owner of the PEM Agent.
<code>--force-registration</code>	Include the <code>force_registration</code> clause to instruct the PEM server to register the agent with the arguments provided; this clause is useful if you are overriding an existing agent configuration. The default value is Yes.
<code>--enable-heartbeat-connection</code>	Enable the <code>enable-heartbeat-connection</code> parameter to create a dedicated heartbeat connection between PEM Agent and server to update the active status. The default value is No.

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 role (`postgres`). 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; you will be prompted for any other required but omitted information. When the registration is complete, the server will confirm that the agent has been successfully registered.

Setting PEM Agent Configuration Parameters

The PEM agent RPM installer creates a sample configuration file named `agent.cfg.sample` in the `/usr/pem/agent/etc` directory. When you register the PEM agent, the `pemworker` program creates the actual agent configuration file (named `agent.cfg`). You must modify the `agent.cfg` file, adding the following configuration parameter:

```
heartbeat_connection = true
```

You must also add the location of the `ca-bundle.crt` file (the certificate authority). By default, the installer creates a `ca-bundle.crt` file in the location specified in your `agent.cfg.sample` file. You can copy the default parameter value from the sample file, or, if you use a `ca-bundle.crt` file that is stored in a different location, specify that value in the `ca_file` parameter:

```
ca_file=/usr/libexec/libcurl-pem7/share/certs/ca-bundle.crt
```

Then, use a platform-specific command to start the PEM agent service; the service is named `pemagent`. For example, on a CentOS or RHEL 6.x system, you would use the command:

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

On a CentOS or RHEL 7.x host, a use `systemctl` to start the service:

```
systemctl start pemagent
```

The service will confirm that it is starting the agent; when the agent is registered and started, it will be displayed on the Global Overview dashboard and in the Object browser tree control of the PEM web interface.

For information about using the `pemworker` utility to register a server, please see the *PEM Getting Started Guide*, available at:

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

3.4 Invoking the Server or Agent Installers from the Command Line

The command line options of the PEM server and PEM agent installers offer functionality in situations where a graphical installation may not work because of limited resources or system configuration. You can:

- Include the `--mode unattended` option when invoking the installer to perform an installation without additional user input.
- Include the `--mode text` option when invoking the installer to perform an installation from the command line with an interactive installer.

Not all command line options are suitable for all platforms. For a complete reference guide to the command line options, include the `--help` option when you invoke the installer.

3.4.1 Invoking the PEM Server Installer in Unattended Mode

You can perform an unattended PEM server installation by providing installation preferences on the command line when invoking the installer. Please note that the system on which you are installing the PEM server must have internet access.

On Linux

Before invoking the installer, you must disable SELinux; for more information, see Section [2.3](#). You must be a superuser to install the PEM server.

The PEM server installer has three dependencies:

- PostgreSQL
- pem-httpd
- Language Pack

You can use the PEM server installer to satisfy the dependencies of the PEM server; use the following command to extract the dependencies:

```
./pem-server-7.x.x-linux-x64.run --extract-dependents /opt
```

After extracting the files, you must install each program. Navigate into the directory that contains the files and enter:

```
./edb-languagepack-version-linux-x64.run --mode unattended
./pem-httpd-version-linux-x64.run --mode unattended
./postgresql-version-linux-x64.run --mode unattended
```

Then, you can invoke the PEM server installer:

```
./pem-server-7.x.x-linux-x64.run --mode unattended
--existing-user registered_edb_user --existing-password
edb_user_password --pgport port --pguser postgres
--pgpassword postgres --cidr-address cidr_address_range
```

Where:

registered_edb_user specifies the name of a registered EnterpriseDB user. To register, visit the EDB website at: <https://www.enterprisedb.com/user/register>

edb_user_password specifies the password associated with the EDB user account.

port specifies the port used by the backing PostgreSQL database; by default, the PostgreSQL database uses port 5432.

cidr_address_range specifies the address range that will be added to the `pg_hba.conf` file of the PEM server's backing database to allow connections from the agents that will be monitored by the server. You may wish to specify a network range (for example, 192.168.2.0/24) to provide server access to agents that reside on the same network.

Please note: when invoked in unattended mode, the PostgreSQL installer creates a user named `postgres`, with a password of `postgres`.

When the installation completes, the installer will notify you that the webservice is installed and the port number that it is monitoring:

```
Configured the webservice for Postgres Enterprise Manager (PEM)
Server on port '8443'
```

On Windows

You must have Administrative privileges to install the PEM server. Before invoking the PEM server installer, you must install the following dependencies:

- PostgreSQL
- pem-httpd
- Language Pack

You can use the PEM server installer to satisfy the dependencies of the PEM server; use the following command to extract the dependencies. Navigate to the location of the installer, and use the following command to extract the dependencies:

```
pem-server-7.x.x-windows-x64.exe --extract-dependents C:\
```

In our example, the files are extracted to the `C:\` directory. After extracting the files, you must install each program. Navigate into the directory that contains the files (in our example, `C:\`), and enter:

```
edb-languagepack-version-windows-x64.exe --mode unattended
pem-httpd-version-windows-x64.exe --mode unattended
postgresql-version-windows-x64.exe --mode unattended
```

Then, you can invoke the PEM server installer:

```
pem-server-7.x.x-windows-x64.exe --mode unattended
--existing-user registered_edb_user --existing-password
edb_user_password --pgport port --pguser postgres
--pgpassword postgres --cidr-address cidr_address_range
--agent_description pem-agent --systempassword windows_password
--agent-crt-path C:\edb
```

Where:

registered_edb_user specifies the name of a registered EnterpriseDB user. To register, visit the EDB website at: <https://www.enterprisedb.com/user/register>

edb_user_password specifies the password associated with the EDB user account.

port specifies the port used by the backing PostgreSQL database; by default, the PostgreSQL database uses port 5432.

cidr_address_range specifies the address range that will be added to the `pg_hba.conf` file of the PEM server's backing database to allow connections from the agents that will be monitored by the server. You may wish to specify a network range (for example, 192.168.2.0/24) to provide server access to agents that reside on the same network.

windows_password specifies the password associated with the Windows Administrator's account.

Please note: when invoked in unattended mode, the PostgreSQL installer creates a user named `postgres`, with a password of `postgres`.

3.4.2 Invoking the PEM Agent Installer in Unattended Mode

You can perform an unattended PEM server installation by providing installation preferences on the command line when invoking the installer. Please note that the system on which you are installing the PEM server must have internet access.

On Linux

Before invoking the PEM agent installer in unattended mode, you must:

- install the PEM server; the `pg_hba.conf` file of the PEM server must allow connections from the host of the PEM agent.
- disable SELinux on the host of the PEM agent; for more information, see Section [2.3](#).
- ensure that the monitored Postgres database has SSL enabled and is accepting connections.

You must be a superuser to install the PEM agent. Use the following command to invoke the PEM agent installer in unattended mode:

```
./pem-agent-7.x.x-linux-x64.run --mode unattended
--pghost pem_server_host_address --pgport pem_server_port
--pguser postgres --pgpassword pguser_password
--agent_description agent_name
```

Where:

pem_server_host_address specifies the IP address of the host of the PEM server.

pem_server_port specifies the port used by the backing PEM database; by default, the database uses port 5432.

pguser_password specifies the password associated with the PEM database superuser.

agent_name specifies a descriptive name for the PEM agent.

On Windows

Before invoking the PEM agent installer in unattended mode, you must:

- install the PEM server; the `pg_hba.conf` file of the PEM server must allow connections from the host of the PEM agent.

- disable SELinux on the host of the PEM agent; for more information, see Section [2.3](#).
- ensure that the monitored Postgres database has SSL enabled, and is accepting connections.

You must have Administrator privileges to install the PEM agent. Use the following command to invoke the PEM agent installer in unattended mode:

```
pem-agent-7.x.x-windows-x64.exe --mode unattended  
--pghost pem_server_host_address --pgport pem_server_port  
--pguser postgres --pgpassword pguser_password  
--agent_description agent_name
```

Where:

pem_server_host_address specifies the IP address of the host of the PEM server.

pem_server_port specifies the port used by the backing PEM database; by default, the database uses port 5432.

pguser_password specifies the password associated with the PEM database superuser.

agent_name specifies a descriptive name for the PEM agent.

3.4.3 Invoking the PEM Server Installer in Text Mode

You can invoke the PEM server installer at the command line to perform an interactive installation if your system does not support a full graphical installation. Please note that the system on which you are installing PEM must have internet access.

Invoking the Installer on Linux

Before invoking the installer, you must disable SELinux; for more information, see Section [2.3](#). You must be a superuser to install the PEM server.

The PEM server installer has three dependencies:

- PostgreSQL
- pem-httpd
- Language Pack

You can use the PEM server installer to satisfy the dependencies of the PEM server; use the following command to extract the dependencies:

```
./pem-server-7.x.x-linux-x64.run --extract-dependents /opt
```

After extracting the files, you must install each program; navigate into the directory that contains the files, and enter:

```
./edb-languagepack-version-linux-x64.run --mode unattended
./pem-httpd-version-1-linux-x64.run --mode unattended
./postgresql-version-linux-x64.run --mode unattended
```

Then, invoke the installer with the command:

```
./pem-server-7.x.x-linux-x64.run --mode text
```

Invoking the Installer on Windows

You must have Administrative privileges to install the PEM server. Before invoking the PEM server installer, you must install the following dependencies:

- PostgreSQL
- pem-httpd
- Language Pack

You can use the PEM server installer to satisfy the dependencies of the PEM server; use the following command to extract the dependencies. Navigate to the location of the installer, and use the following command to extract the dependencies:

```
pem-server-7.x.x-windows-x64.exe --extract-dependents C:\
```

After extracting the files, you must install each program; navigate into the directory that contains the files, and enter:

```
edb-languagepack-version-windows-x64.exe --mode unattended
pem-httpd-version-windows-x64.exe --mode unattended
postgresql-version-windows-x64.exe --mode unattended
```

Then, you can invoke the PEM server installer with the following command:

```
pem-server-7.x.x-windows-x64.exe --mode text
```

Example

The example that follows demonstrates performing a text-mode installation on a Linux system; the installation process on a Windows system is similar.

When you invoke the installer, the installer welcomes you:

```
-----
Welcome to the Postgres Enterprise Manager (PEM) Server Setup
Wizard.
-----
```

Before installing the PEM server, you must review and accept the terms of the PEM license agreement:

```
Please read the following License Agreement. You must accept the
terms of this agreement before continuing with the installation.
Review the license agreement; when prompted, accept the
agreement, and press Return.
Do you accept this license? [y/n]: y
-----
```

After accepting the license agreement, you will be prompted for the name and address of a registered EnterpriseDB user; provide the information when prompted:

```
EnterpriseDB User Account Information
```

```
Please enter the email address and password for your
enterprisedb.com user account.
```

Email address []:

Password :

Next, you will be prompted for an installation directory; you can use the default installation directory, or specify an alternate location.

Enter a Y after Show advanced options to access menu options that allow you to use a pre-existing Postgres server installation or PEM-HTTPD installation with the PEM server.

Installation Directory

Please select a directory for the PEM server installation.

Installation Directory [/opt/edb/pem]:

Show advanced options [y/N]:

Use the fields that follow the Database Server Installation Details heading to provide the name and password of the PostgreSQL database superuser for the PEM backing database:

Database Server Installation Details

Please specify user and password for the local server - 'PostgreSQL 10' installation running on port 5432.

User [postgres]:

Password :

When prompted, provide the network address or range that will be added to the pg_hba.conf file of the PEM server host; this will allow PEM agents on monitored hosts to contact the server:

Network Details

Please enter the CIDR formatted network address range that agents will connect to the server from, to be added to the server's pg_hba.conf file. For example, 192.168.1.0/24.

Network address [127.0.0.1/32]:

Use the Agent Details fields to specify a descriptive name for the agent and the location of the agent certificate:

Agent Details.

Please specify a description for the agent.

Description. [Postgres Enterprise Manager Host]:

Agent certificate path [/root/.pem]:

The installer will prompt you before it proceeds with the installation; press Return to start the installation:

```
Setup is now ready to begin installing the PEM server on your
computer.
Do you want to continue? [Y/n]:
Please wait while Setup installs the PEM server on your computer.
```

```
Installing
0% _____ 50% _____ 100%
#####
```

The installer will notify you when the installation is complete:

```
Info: Configured the webservice for Postgres Enterprise Manager
(PEM) Server on port '8443'.
Created and configured the 'pem' database.
EnterpriseDB is the leading provider of value-added products and
services for the Postgres community.
Please visit our website at www.enterprisedb.com
```

3.4.4 Invoking the PEM Agent Installer in Text Mode

You can invoke the PEM agent installer at the command line to perform an interactive installation if your system does not support a full graphical installation. Please note that the system on which you are installing the agent must have access to the PEM server.

Invoking the Installer on Linux

Before invoking the installer, you must disable SELinux; for more information, see Section [2.3](#). You must be a superuser to install the PEM agent.

Invoke the installer with the command:

```
./pem-agent-7.x.x-linux-x64.run --mode text
```

Invoking the Installer on Windows

You must have Administrative privileges to install the PEM server. You can invoke the PEM server installer with the following command:

```
pem-server-7.x.x-windows-x64.exe --mode text
```

Example

The example that follows demonstrates performing a text-mode installation on a Linux system; the installation process on a Windows system is similar.

When you invoke the PEM agent installer, the installer welcomes you:

```
-----
Welcome to the Postgres Enterprise Manager (PEM) Agent Setup Wizard.
-----
```

Before installing the PEM server, you must review and accept the terms of the PEM license agreement:

```
Please read the following License Agreement. You must accept the
terms of this agreement before continuing with the installation.
```

Press [Enter] to continue:
Do you accept this license? [y/n]:

Next, you will be prompted for an installation directory; you can use the default installation directory, or specify an alternate location.

By default, the PEM agent installer places a certificate in `~root/.pem` (on Linux) or in the Administrator's `%APPDATA%\pem` directory (on Windows). Enter a Y after Show advanced options to access menu options that allow you to specify an alternate path for the certificate file.

Installation Directory

Please select a directory for PEM agent installation.
Installation Directory [/opt/edb/pem]:
Show advanced options [y/N]:

When prompted, provide information about the PEM server installation:

PEM server installation details

Please verify the PEM server installation details
Host [localhost]:
User Name [postgres]:
Password :
Port [5432]:

You can provide a descriptive name for the agent, or press return to accept the default:

Agent Details

Please provide the agent description
Description [localhost]:

The installer will prompt you before it proceeds with the installation; press Return to start the installation:

Setup is now ready to begin installing the PEM agent on your computer.
Do you want to continue? [Y/n]:

Please wait while Setup installs the PEM agent on your computer.

EDB Postgres Enterprise Manager Installation Guide

Installing

0% _____ 50% _____ 100%

#####

The installer will notify you when the installation is complete:

EnterpriseDB is the leading provider of value-added products and services for the Postgres community.

Please visit our website at www.enterprisedb.com

4 The PEM Web Interface

After installing a PEM server and agent, you can configure PEM to start monitoring and managing PostgreSQL or Advanced Server instances. The PEM server installer installs the PEM web interface. You can use the interface to review information about objects that reside on monitored servers, or to review statistical information gathered by the PEM server.

After installing and configuring PEM, you can use your browser to access the PEM web interface. Open your browser, and navigate 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 Web Login window (shown in Figure 4.1) opens:

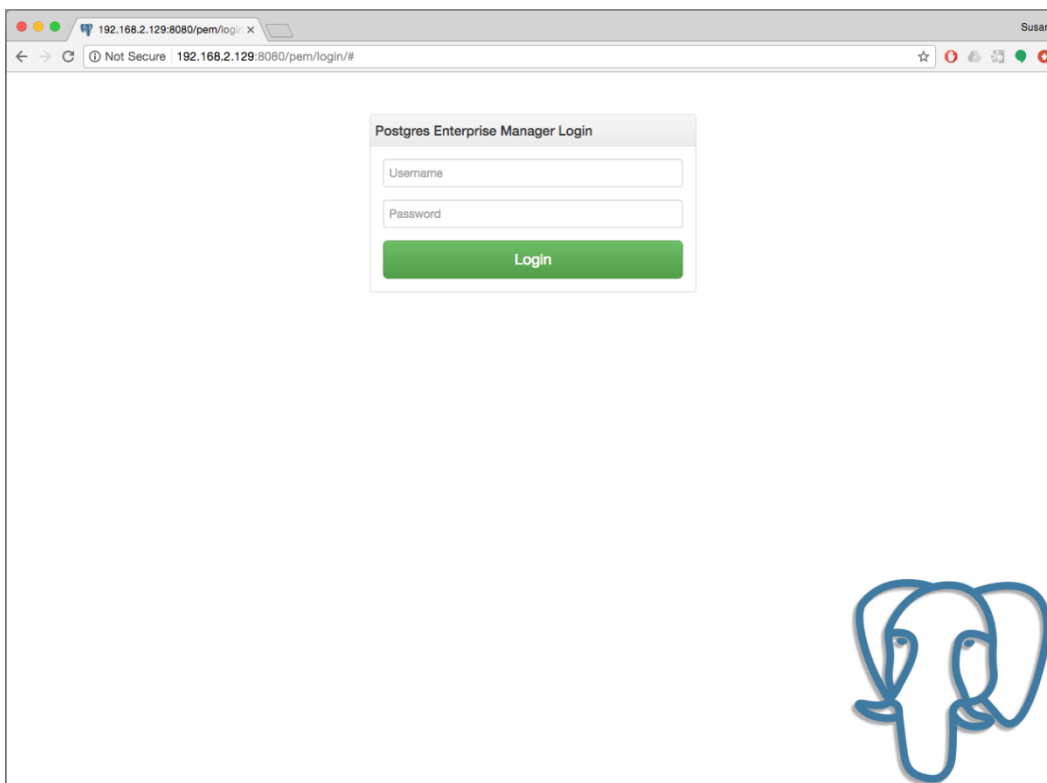


Figure 4.1 - The PEM Web Login page.

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.

Click the `Login` button to connect to the PEM server.

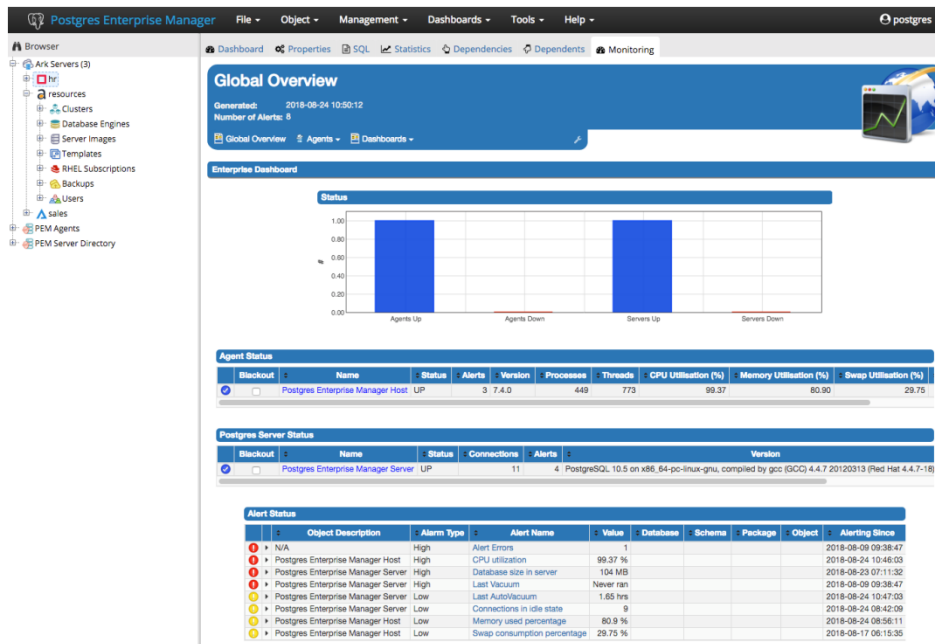


Figure 4.2 - The PEM web interface.

Before you can use the PEM web interface to manage or monitor a database server, you must *register* the server with the PEM server. When you register a server, you describe the connection to the server, provide authentication information for the connection, and specify any management preferences (optionally binding an agent).

A server may be managed or unmanaged:

- A *managed* server is bound to a PEM agent. The PEM agent will monitor the server to which it is bound, and perform tasks or report statistics for display on the PEM dashboards. A managed server has access to extended PEM functionality such as Package Management or Custom Alerting; when registering a server, you can also allow a managed server to be restarted by PEM as required.
- An *unmanaged* server is not bound to a PEM agent; you can create database objects on an unmanaged server, but extended PEM functionality (such as Package Management or Custom Alerting) is not supported on an unmanaged server.

You must also ensure the `pg_hba.conf` file of the server that you are registering allows connections from the host of the PEM web interface.

To access online help information about the PEM web interface, select `Help` from the menu bar (see Figure 4.2). Additional information is available in `.pdf` and `.html` format from the EnterpriseDB website at:

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

The *PEM Getting Started Guide* contains information about registering and managing servers, agents, and users.

The *PEM Enterprise Features Guide* contains information about using the tools and wizards that are part of the web interface.

The *PEM EDB Ark Management Guide* contains information about using PEM to manage cloud installations of Advanced Server and PostgreSQL.

The *PEM Upgrade and Migration Guide* contains information about upgrading to PEM 7.6 from a previous version.

5 Installing the SQL Profiler Plugin

The SQL Profiler Plugin allows you to profile a server's workload. You must install the plugin on each server on which you wish to use SQL Profiler. For example, if you have a host running PostgreSQL 9.6 and PostgreSQL 10, you must install two versions of the plugin, one for each server.

SQL Profiler for Postgres Enterprise Manager™ is officially supported only on the EnterpriseDB distributions of PostgreSQL version 9.4 (and above) and Advanced Server version 9.4 (and above). The plugin is distributed via StackBuilder, or is available from the EnterpriseDB website at:

<https://www.enterprisedb.com/advanced-downloads>

The plugin is also distributed and installed with the server component of the Advanced Server installer.

The SQL Profiler plugin may be installed on servers with or without a PEM agent, however traces can only be run in ad-hoc mode on unmanaged servers, and may only be scheduled on managed servers.

Follow the installation steps listed below to install the plugin for PostgreSQL before continuing to the `Configuration` section. If you are using Advanced Server, skip ahead to the `Configuration` section.

5.1 Installing SQL Profiler

To invoke the SQL Profiler installer, assume superuser privileges (or Administrator privileges on Windows), navigate into the directory that contains the installer, and invoke the installer with the command:

```
sqlprofiler-pg-x.x-6.0.x-platform.ext
```

The SQL Profiler installer welcomes you to the Setup Wizard (as shown in Figure 5.1).



Figure 5.1 - The SQL Profiler Installer Welcome dialog.

Click **Next** to continue to the License Agreement, shown in Figure 5.2.

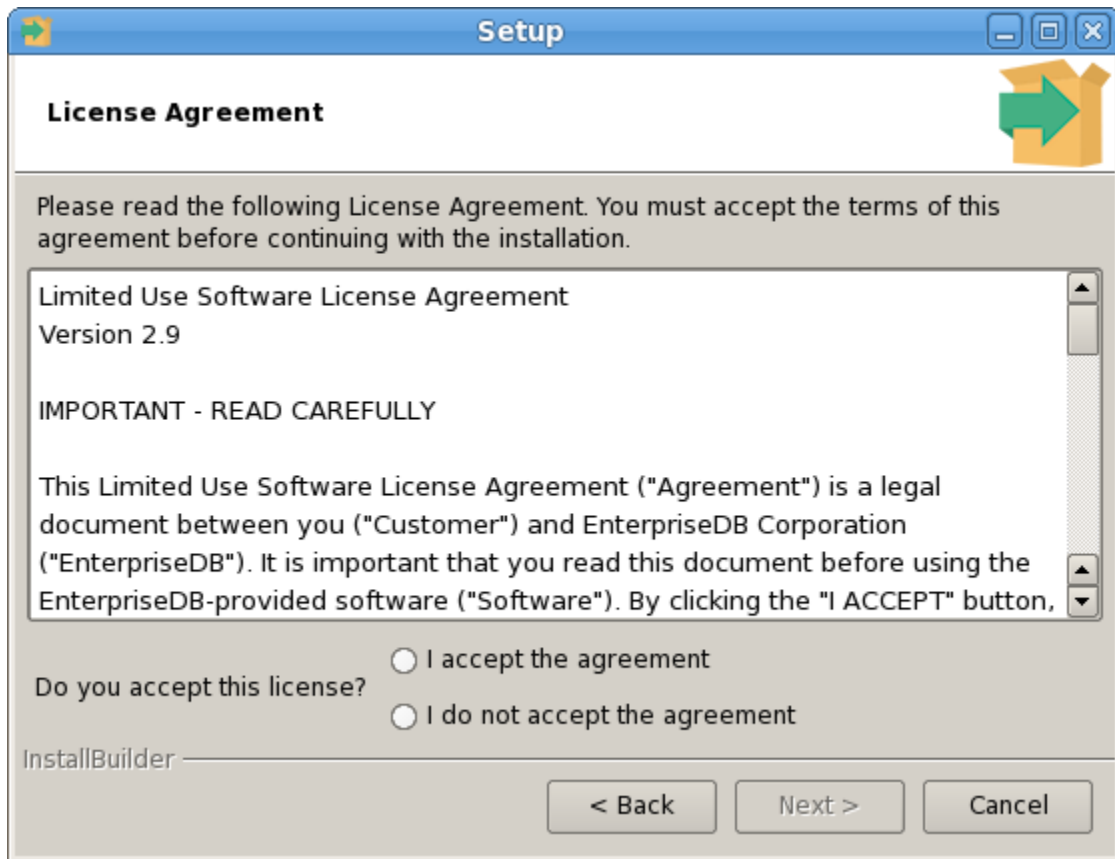


Figure 5.2 - The SQL Profiler License Agreement.

Carefully review the license agreement before highlighting the appropriate radio button and accepting the agreement; click `Next` to continue to the `Installation Directory` dialog (shown in Figure 5.3).

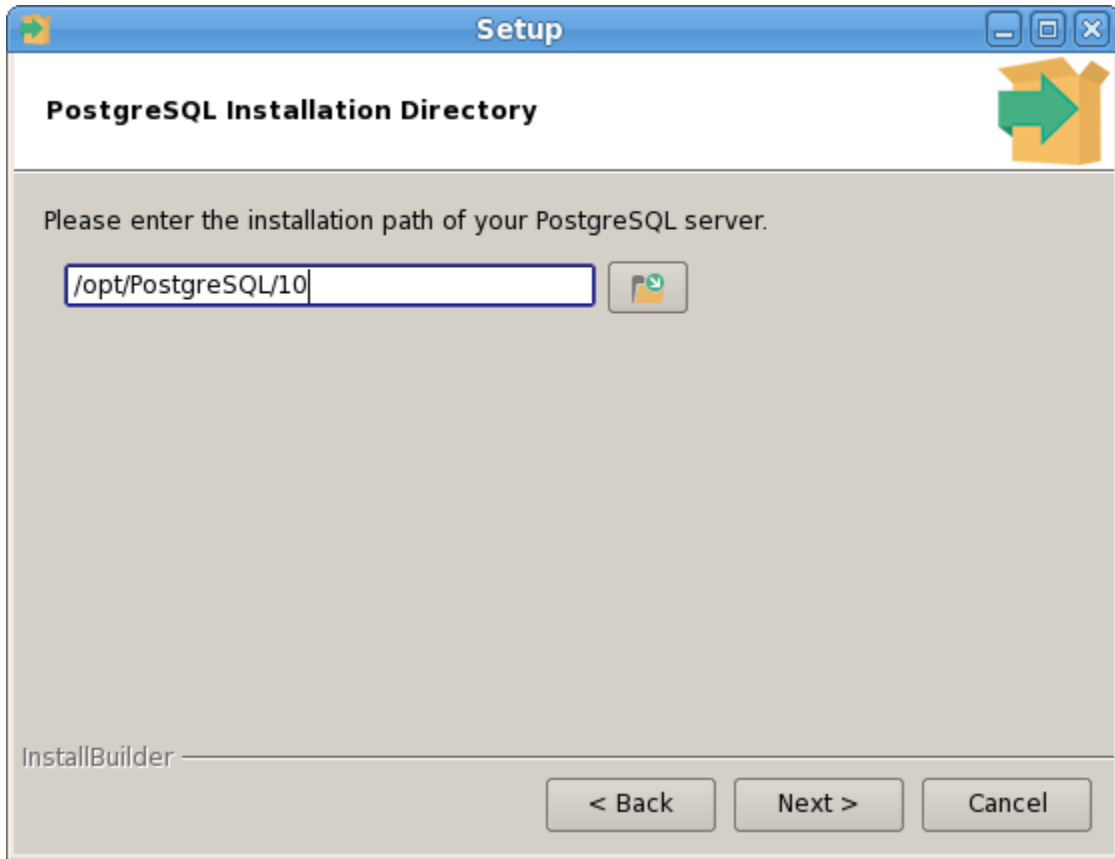


Figure 5.3 - The PostgreSQL Installation Directory

Specify an alternate location for the installation directory, or accept the default location and click `Next` to continue.

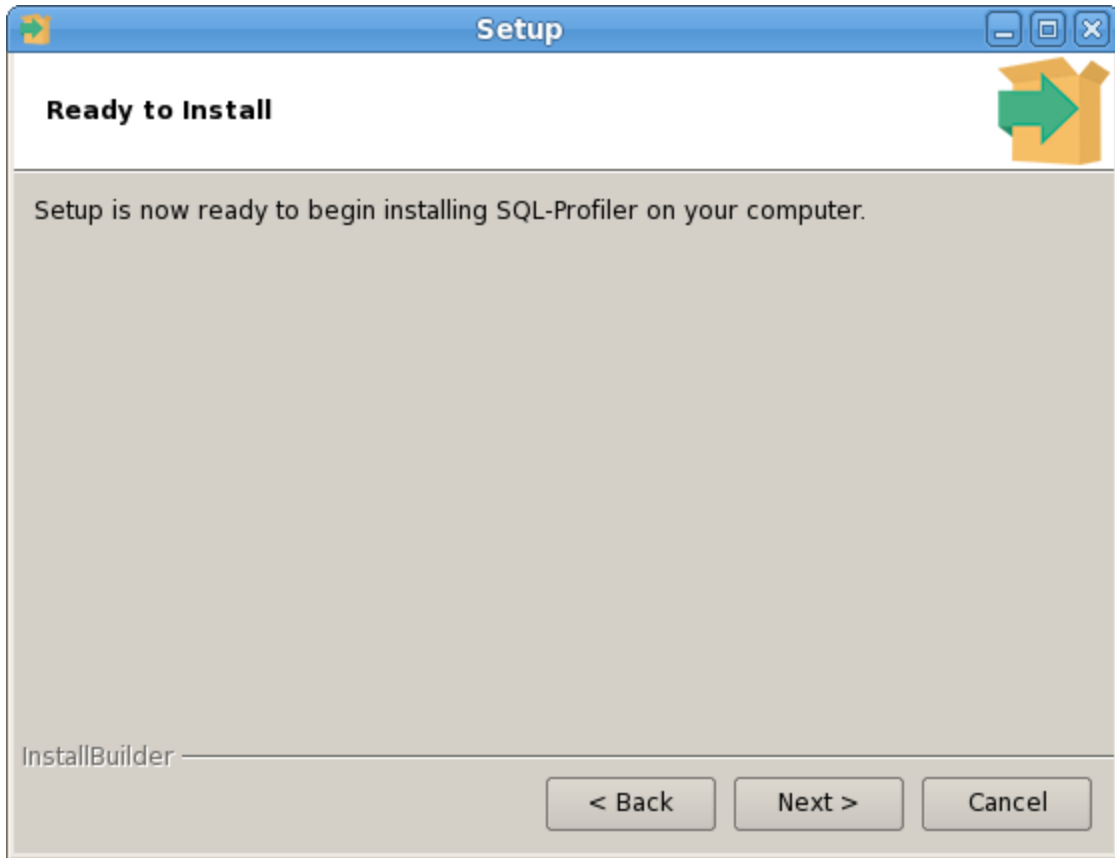


Figure 5.4 - Ready to Install

The wizard is now ready to install the SQL Profiler plugin (as shown in Figure 5.4). Click `Next` to continue.

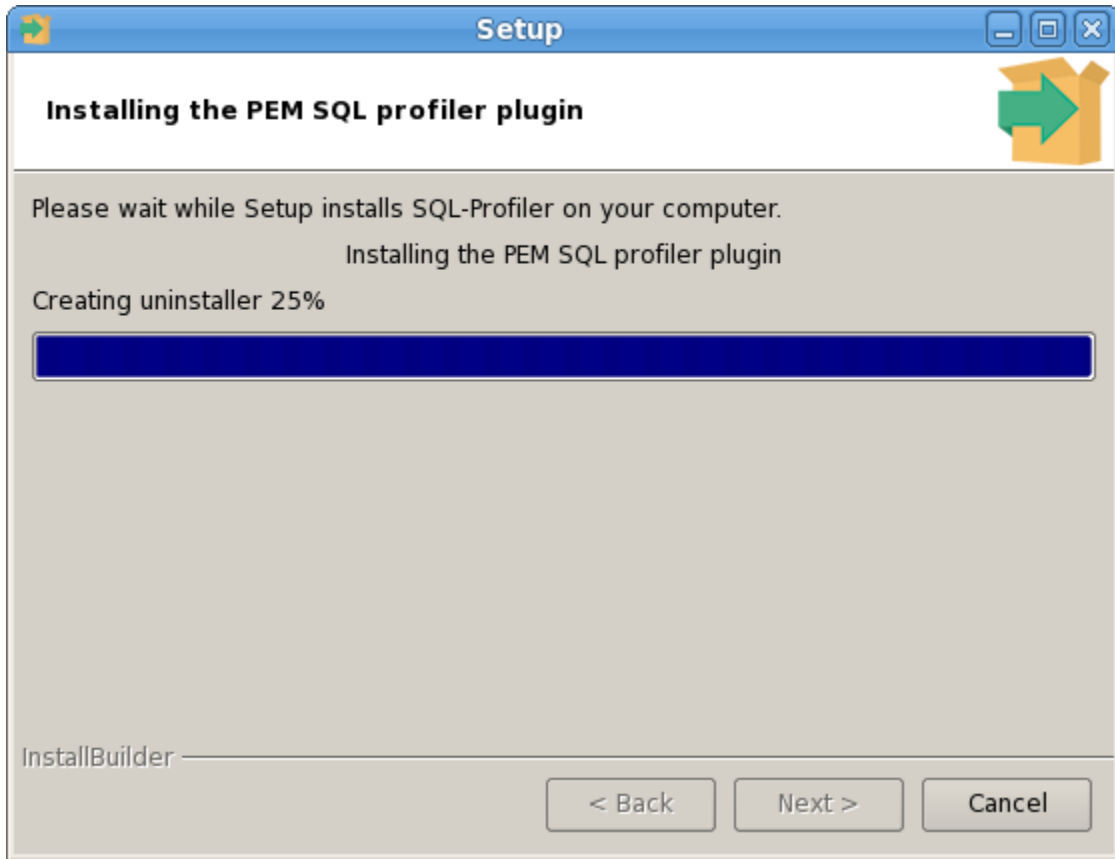


Figure 5.5 - Installing the SQL Profiler plugin.

The SQL Profiler plugin installer displays progress bars as it copies files to your system (see Figure 5.5).

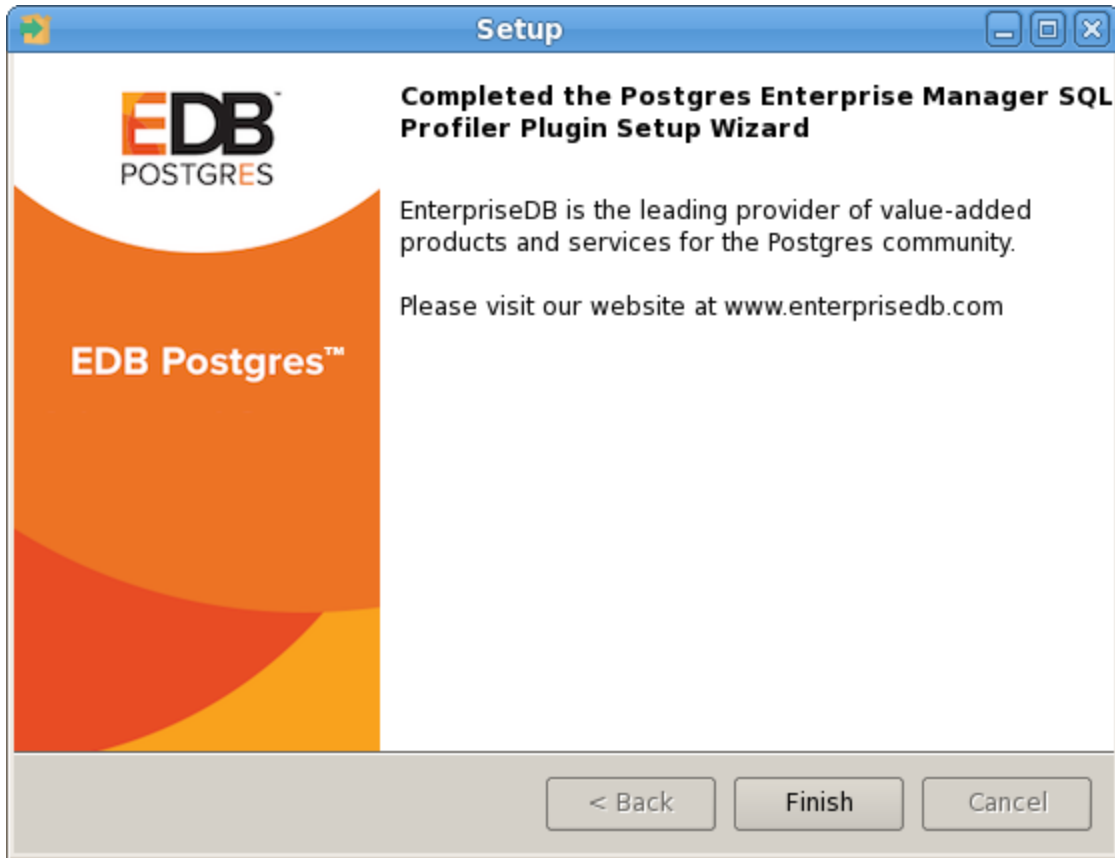


Figure 5.6 - The SQL Profiler installation is complete.

When the installation is complete, the SQL Profiler plugin is ready to be configured.

5.2 Configuring SQL Profiler

The SQL Profiler plugin is not automatically enabled when the installation process completes. This allows you to restart the server at a convenient time, and prevents the plugin from being loaded unnecessarily on systems where it is not required on a continual basis.

Use the following steps to enable the plugin:

1. Edit the `postgresql.conf` file on the server you wish to profile, modifying the `shared_preload_libraries` parameter as shown below:

```
shared_preload_libraries = '$libdir/sql-profiler'
```

2. Restart the Postgres server.
3. Using the Query Tool or the `psql` command line interface, run the `sql-profiler.sql` script in the database specified as the Maintenance Database on the server you wish to profile. If you are using:

- PostgreSQL, the default maintenance database is `postgres`.
- Advanced Server, the default maintenance database is `edb`.

To use the PEM Query Tool to run the script, highlight the name of the maintenance database in the `Browser` tree control, and navigate through the `Tools` menu to select `Query tool`. When the Query Tool opens, use the `Open` option on the `Files` menu to open a web browser and navigate to the `sql-profiler.sql` script. By default, the `sql-profiler.sql` script is located in the `contrib` folder, under your Postgres installation.

When the script opens in the `SQL Editor` panel of the Query Tool, highlight the content of the script in the `SQL Editor` and select the `Execute` option from the `Query` menu (or click the `Execute` icon) to invoke the script and configure SQL Profiler.

You can also use the `psql` command line to invoke the configuration script. The following command uses `psql` to invoke the `sql-profiler.sql` script on an Advanced Server database on a Linux system:

```
$ /opt/edb/as10/bin/psql -U postgres postgres <
/opt/edb/as10/share/contrib/sql-profiler.sql
```

After configuring SQL Profiler, it is ready to use with all databases that reside on the server.

To access SQL Profiler functionality, highlight the name of the database in the PEM Browser tree control; navigate through the Management menu to the SQL Profiler pull-aside menu. Menu options allow you to manage your SQL traces:

- Select `Create trace...` to define a new trace (see Figure 5.7).
- Select `Open trace...` to open an existing trace.
- Select `Delete trace(s)...` to delete one or more traces.
- Select `View scheduled trace(s)...` to review a list of scheduled traces.

The screenshot shows the 'Create trace...' dialog box. It has a blue title bar with the text 'Create trace...' and window control icons. Below the title bar are three tabs: 'Trace options' (selected), 'Schedule', and 'Periodic job options'. The 'Trace options' section contains the following fields and controls:

- Name:** An empty text input field.
- User filter:** A dropdown menu with the text 'Select the users...'.
- Database filter:** A dropdown menu with the text 'Select the databases...'.
- Maximum trace file size(MB):** A spinner control showing the value '0'. Below it is a note: 'Note: 0(Zero) indicates maximum trace file size to be 250 MB'.
- Run now?:** A checkbox that is checked, with a green 'Yes' button next to it.

At the bottom of the dialog, there is a red error bar with the text 'Please provide valid name for trace'. In the bottom right corner, there are two buttons: 'Create' (with a document icon) and 'Cancel' (with an 'X' icon).

Figure 5.7 – Creating a new trace.

For more information about using SQL Profiler, consult the online help text for SQL Profiler (accessed through the `Help` menu) or the PEM Getting Started Guide.

6 Upgrading and Uninstalling PEM Components

For detailed information about upgrading and moving PEM components, please see the PEM Upgrade and Migration Guide. PEM documentation and links to PEM and SQL Profiler installers are available from the EnterpriseDB website at:

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

Upgrading Postgres Enterprise Manager™

To upgrade your version of PEM, simply invoke the PEM component installers. Each installer will notify you if it detects a previous installation, and will upgrade to the more recent version of PEM.

You should invoke the PEM installers in the following order:

1. Invoke the PEM agent installer on each monitored node *except* the PEM server node.
2. Invoke the PEM server installer; this installer will upgrade *both* the PEM server and the PEM agent that resides on the PEM server node.

For detailed information about upgrading from a previous version, see the *PEM Upgrade and Migration Guide*.

Upgrading SQL Profiler on a Linux Host

To upgrade a SQL Profiler installation that resides on a Linux host:

1. Delete the existing SQL Profiler query set on each node by invoking the `uninstall-sql-profiler.sql` script.

By default, if you are using Advanced Server on a Linux host, the script resides in:

```
/opt/PostgresPlus/9.x/share/contrib
```

If you are using a PostgreSQL installation on a Linux host, the script resides in:

```
/opt/PostgreSQL/9.x/share/postgresql/contrib
```

Where `x` specifies the installed Postgres version.

2. Then, invoke the new SQL Profiler installer on each node you wish to profile.

Upgrading SQL Profiler on a Windows Host

If you are using SQL Profiler on a Windows host, Windows will lock any files that have been executed or loaded into memory. To release any locked files, you must stop the Postgres server before performing an upgrade.

After stopping the Postgres Server:

1. Delete the existing SQL Profiler query set on each node by invoking the `uninstall-sql-profiler.sql` script.

If you are using Advanced Server on a Windows host, the script resides in:

```
C:\Program Files\PostgresPlus\9.x\share\contrib
```

If you are using a PostgreSQL installation on a Windows host, the script resides in:

```
C:\Program Files\PostgreSQL\9.x\share\contrib
```

Where `x` specifies the installed Postgres version.

2. Invoke the new SQL Profiler installer on each node you wish to profile.

Then, restart the Postgres Server, to resume profiling the node.

Troubleshooting

If (after upgrading to a newer version of SQL Profiler) you encounter the following error:

```
An error has occurred:
ERROR: function return row and query-specified return row
do not match
DETAIL: Returned row contains 11 attributes, but query
expects 10.
```

To correct this error, you must replace the existing query set with a new query set. First, uninstall SQL Profiler by invoking the `uninstall-sql-profiler.sql` script, and then reinstall SQL Profiler by invoking the `sql-profiler.sql` script.

Uninstalling Postgres Enterprise Manager™

Use the uninstallers provided in the PEM installation directory to remove PEM components from an Advanced Server or PostgreSQL installation on Linux or Solaris. By default, the PEM uninstallers are located:

Component	Uninstaller name	Default location
PEM server	uninstall-pemserver	/opt/PEM/server
PEM agent	uninstall-pemagent	/opt/PEM/agent
SQL Profiler	uninstall-sqlprofiler	/opt/PostgreSQL/10

To remove a component, assume superuser privileges, open a terminal window, and navigate into the directory in which the uninstaller resides; invoke the installer as follows:

```
./uninstall-component_name
```

Where *component_name* is the name of the component that you wish to remove.

If the PEM installation resides on a Windows host, you can use the Windows Uninstall a Program applet to remove PEM components. To open the Uninstall a Program applet, navigate through the Programs submenu on the Windows Control Panel, selecting Programs and Features. When the Uninstall a Program window opens, highlight the name of the PEM component that you wish to remove, and click the Uninstall/Change button. A Windows popup will open, prompting you to confirm that you wish to remove the component; click Yes to remove the component.

7 Reference - Linux Service Script

The Postgres server on which the PEM server resides must contain a service script. Postgres installers generated by EnterpriseDB create a service script for you; if you are using a Postgres server from another source, you must provide a service script.

You can use the following example of a linux service script as a starting point when developing a script for a Postgres installation that was installed or built from a source that does not provide one. Please ensure (if you copy and paste from this example) that the line breaks are copied correctly.

```
#!/bin/bash
#
# chkconfig: 2345 85 15
# description: Starts and stops the PostgreSQL/Postgres Plus
Advanced Server database server

# PostgreSQL/Postgres Plus Advanced Server Service script
template for Linux
# Please modify the values accordingly

DB_DESC="Database Server - PostgreSQL 9.5"
DB_INSTALL_DIR=/opt/PostgreSQL/9.5
DB_BIN_DIR=${DB_INSTALL_DIR}/bin
DB_LIB_DIR=${DB_INSTALL_DIR}/lib
DB_DATA_DIR=${DB_INSTALL_DIR}/data
DB_HBA_FILE=${DB_DATA_DIR}/pg_hba.conf
DB_CONF_FILE=${DB_DATA_DIR}/postgresql.conf
DB_PID_FILE=${DB_DATA_DIR}/postmaster.pid
DB_STARTUP_LOG=${DB_DATA_DIR}/pg_log/startup.log
DB_SERVICE_USER=postgres

_die()
{
    echo ""
    echo "FATAL ERROR: $*"
    echo ""
    exit 1
}

if [ `id -u` != 0 ]; then
    _die "You must run this script as the root."
fi

# Source function library.
if [ -f /etc/rc.d/functions ];
then
    . /etc/init.d/functions
fi
```

```

start()
{
    STARTDBSERVER=0
    if [ -e "${DB_PID_FILE}" ]
    then
        PIDOFDB=`head -n 1 "${DB_PID_FILE}"`
        PIDALIVEDB=""
        if [ -n "${DB_PID_FILE}" ]; then
            PIDALIVEDB=`ps -p "${PIDOFDB}" | grep
"${PIDOFDB}"`
        fi

        if [ -n "${PIDALIVEDB}" ]
        then
            echo "The '${DB_DESC}' is already running.
PID(${PIDALIVEDB})."
            exit
        else
            STARTDBSERVER=1
        fi
    else
        STARTDBSERVER=1
    fi

    if [ "${STARTDBSERVER}" != 0 ]
    then
        echo "Starting ${DB_DESC}..."
        su - "${DB_SERVICE_USER}" -c
"LD_LIBRARY_PATH=\"${DB_LIB_DIR}:\$LD_LIBRARY_PATH\"
\"${DB_BIN_DIR}/pg_ctl\" -w start -D \"${DB_DATA_DIR}\" -l
\"${DB_STARTUP_LOG}\" -o \"${DB_STARTUP_OPTIONS}\""

        if [ $? -eq 0 ];
        then
            echo "${DB_DESC} started successfully."
            exit 0
        else
            echo "${DB_DESC} did not start in a timely
fashion, please see '${DB_STARTUP_LOG}' for details."
            exit 1
        fi
    fi
}

stop()
{
    if [ -e "${DB_PID_FILE}" ]
    then
        PIDOFDB=`head -n 1 "${DB_PID_FILE}"`
        PIDALIVEDB=""
        if [ -n "${DB_PID_FILE}" ]; then

```

EDB Postgres Enterprise Manager Installation Guide

```

        PIDALIVEDB=`ps -p "${PIDOFDB}" | grep
"${PIDOFDB}"`
        fi

        if [ -n "${PIDALIVEDB}" ]
        then
            echo "Stopping ${DB_DESC}..."
            su - "${DB_SERVICE_USER}" -c
"LD_LIBRARY_PATH=\"${DB_LIB_DIR}:\$LD_LIBRARY_PATH\"
\"${DB_BIN_DIR}/pg_ctl\" stop -m fast -D \"${DB_DATA_DIR}\" -l
\"${DB_STARTUP_LOG}\" -o \"${DB_STARTUP_OPTIONS}\""
            else
                echo "The '${DB_DESC}' is not running."
            fi
        else
            echo "The '${DB_DESC}' is not running."
        fi
    }

reload()
{
    echo "Reloading '${DB_DESC}'..."
    su - "${DB_SERVICE_USER}" -c
"LD_LIBRARY_PATH=\"${DB_LIB_DIR}:\$LD_LIBRARY_PATH\"
\"${DB_BIN_DIR}/pg_ctl\" reload -D \"${DB_DATA_DIR}\" -l
\"${DB_STARTUP_LOG}\" -o \"${DB_STARTUP_OPTIONS}\""
}

restart()
{
    echo "Restarting '${DB_DESC}'..."
    su - "${DB_SERVICE_USER}" -c
"LD_LIBRARY_PATH=\"${DB_LIB_DIR}:\$LD_LIBRARY_PATH\"
\"${DB_BIN_DIR}/pg_ctl\" restart -m fast -w -D \"${DB_DATA_DIR}\"
-l \"${DB_STARTUP_LOG}\" -o \"${DB_STARTUP_OPTIONS}\""

    if [ $? -eq 0 ];
    then
        echo "'${DB_DESC}' restarted successfully."
        exit 0
    else
        echo "'${DB_DESC}' did not start in a timely fashion,
please see '${DB_STARTUP_LOG}' for details."
        exit 1
    fi
}

_die_incomplete_requirement()
{
    echo "One or more required configuration variables are not
set:"
    _die $*
}

```



```

}

_validate_script()
{
    if [ -z "${DB_INSTALL_DIR}" ]; then
        _die_incomplete_requirement "Missing installation
directory";
    fi
    if [ ! -d "${DB_INSTALL_DIR}" ]; then
        _die_incomplete_requirement "The specified -
'${DB_INSTALL_DIR}' is not a valid installation directory. It is
not present on the system.";
    fi

    if [ -z "${DB_BIN_DIR}" ]; then
        DB_BIN_DIR=${DB_INSTALL_DIR}/bin; fi
    if [ ! -d "${DB_BIN_DIR}" ]; then
        _die_incomplete_requirement "The specified -
'${DB_BIN_DIR}' is not a valid bin directory. It is not present
on the system.";
    fi
    if [ ! -f "${DB_BIN_DIR}/pg_config" -o ! -f
"${DB_BIN_DIR}/pg_ctl" ]; then
        _die_incomplete_requirement "The specified -
'${DB_BIN_DIR}' does not contain the database server binaries.";
    fi

    if [ -z "${DB_LIB_DIR}" ]; then
        DB_LIB_DIR=${DB_INSTALL_DIR}/lib; fi
    if [ -z "${DB_DESC}" ]; then
        DB_DESC=`${DB_BIN_DIR}/pg_config --version`; fi

    if [ -z "${DB_DATA_DIR}" ]; then
        _die_incomplete_requirement "Missing data directory
settings in the script. Please set 'DB_DATA_DIR' variable in the
script.";
    fi

    if [ ! -d "${DB_DATA_DIR}" ]; then
        _die_incomplete_requirement "The specified -
'${DB_DATA_DIR}' is not a valid. It is not present on the
system.";
    fi
    if [ ! -f "${DB_DATA_DIR}/PG_VERSION" -o ! -d
"${DB_DATA_DIR}/base" -o ! -d "${DB_DATA_DIR}/global" ]; then
        _die_incomplete_requirement "The directory -
'${DB_DATA_DIR}' does not look like a valid PostgreSQL/Postgres
Plus Advanced Server data directory."
    fi
    if [ -z "${DB_SERVICE_USER}" ]; then
        _die_incomplete_requirement "The service-user is not
specified in the service script. Please set 'DB_SERVICE_USER'

```

```

variable in the script."
    fi
    DB_VALID_SERVICE_USER=`cat /etc/passwd | grep
"^${DB_SERVICE_USER}:"`
    if [ -z "${DB_VALID_SERVICE_USER}" ]; then
        _die_incomplete_requirement "The service-user
'${DB_SERVICE_USER}' is not present on the system. Please specify
the correct information."
    fi
    DB_DATA_DIR_OWNER=`ls -l ${DB_DATA_DIR}/PG_VERSION | awk
'{print $3}'`
    if [ x"${DB_DATA_DIR_OWNER}" != x"${DB_SERVICE_USER}" ];
then
        _die_incomplete_requirement "The specified user -
'${DB_SERVICE_USER}' does not own the data directory -
'${DB_DATA_DIR}'. The data directory is owned by the user -
'${DB_DATA_DIR_OWNER}'."
    fi
    if [ -z "${DB_HBA_FILE}" ]; then
DB_HBA_FILE=${DB_DATA_DIR}/pg_hba.conf; fi
    if [ ! -f "${DB_HBA_FILE}" ]; then
        _die_incomplete_requirement "The hba-file -
'${DB_HBA_FILE}' does not exist."
    fi
    if [ -z "${DB_CONF_FILE}" ]; then
DB_CONF_FILE=${DB_DATA_DIR}/postgresql.conf; fi
    if [ ! -f "${DB_CONF_FILE}" ]; then
        _die_incomplete_requirement "The config-file -
'${DB_CONF_FILE}' does not exist."
    fi
    if [ -z "${DB_PID_FILE}" ]; then
DB_PID_FILE=${DB_DATA_DIR}/postmaster.pid; fi
    if [ -z "${DB_STARTUP_LOG}" ]; then
DB_STARTUP_LOG=${DB_DATA_DIR}/pg_log/startup.log; fi

    DB_STARTUP_OPTIONS=""
    if [ x"${DB_CONF_FILE}" !=
x"${DB_DATA_DIR}/postgresql.conf" ]; then
        DB_STARTUP_OPTIONS="-c 'config_file=${DB_CONF_FILE}'"
    fi
    if [ x"${DB_HBA_FILE}" != x"${DB_DATA_DIR}/pg_hba.conf" ];
then
        DB_STARTUP_OPTIONS="${DB_STARTUP_OPTIONS} -c
'hba_file=${DB_HBA_FILE}'"

    fi

    if [ x"${DB_PID_FILE}" != x"${DB_DATA_DIR}/postmaster.pid"
]; then

```

```

        DB_STARTUP_OPTIONS="${DB_STARTUP_OPTIONS} -c
external_pid_file=${DB_PID_FILE}"
    fi

    if [ x"${DEBUG_VALIDATION}" = x"1" ]; then
        echo "Using these values in the scripts:"
        echo ""
        echo "DB_DESC           : ${DB_DESC}"
        echo ""
        echo "DB_INSTALL_DIR      : ${DB_INSTALL_DIR}"
        echo "DB_BIN_DIR          : ${DB_BIN_DIR}"
        echo "DB_LIB_DIR          : ${DB_LIB_DIR}"
        echo ""
        echo "DB_DATA_DIR         : ${DB_DATA_DIR}"
        echo "DB_HBA_FILE         : ${DB_HBA_FILE}"
        echo "DB_CONF_FILE        : ${DB_CONF_FILE}"
        echo "DB_PID_FILE         : ${DB_PID_FILE}"
        echo "DB_STARTUP_LOG      : ${DB_STARTUP_LOG}"
        echo ""
        echo "DB_SERVICE_USER     : ${DB_SERVICE_USER}"
        echo "DB_STARTUP_OPTIONS  : ${DB_STARTUP_OPTIONS}"
        echo ""
    fi
}

DEBUG_VALIDATION=0
# See how we were called.
case "$1" in
    start)
        _validate_script
        start
        ;;
    stop)
        _validate_script
        stop
        ;;
    reload)
        _validate_script
        reload
        ;;
    restart)
        _validate_script
        restart
        ;;
    condrestart)
        _validate_script
        if [ -e "${DB_PID_FILE}" ]
        then
            PIDOFDB=`head -n 1 "${DB_PID_FILE}"`
            PIDALIVEDB=""
            if [ -n "${DB_PID_FILE}" ]; then
                PIDALIVEDB=`ps -p "${PIDOFDB}" | grep

```

```

"${PIDOFDB}"`
    fi
    if [ -n "${PIDALIVEDB}" ]
    then
        restart
    else
        echo "The '${DB_DESC}' is not running."
    fi
else
    echo "The '${DB_DESC}' is not running."
fi
;;
status)
    _validate_script
    su - "${DB_SERVICE_USER}" -c
"LD_LIBRARY_PATH=\"${DB_LIB_DIR}:\$LD_LIBRARY_PATH\"
\"${DB_BIN_DIR}/pg_ctl\" status -D \"${DB_DATA_DIR}\" -l
\"${DB_STARTUP_LOG}\" -o \"${DB_STARTUP_OPTIONS}\""
    ;;
validate)
    DEBUG_VALIDATION=1
    _validate_script
    exit 0
    ;;
*)
    echo "Usage: $0
{start|stop|restart|condrestart|reload|status|validate}"
    exit 1
esac

```