# Table of Contents

1 Introduction .................................................................................................................. 5
  1.1 Typographical Conventions Used in this Guide ....................................................... 6

2 Postgres Enterprise Manager™ - Overview ..................................................................... 7
  2.1 Supported Platforms ................................................................................................. 9
  2.2 Pre-Requisites .......................................................................................................... 10

3 Installing Postgres Enterprise Manager™ ..................................................................... 11
  3.1 Installing the PEM Server ........................................................................................ 12
    3.1.1 Installing the PEM Server and Apache/PHP on the Same Host ...................... 14
    3.1.2 Installing the PEM Server and Apache/PHP 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 ......................... 44
      3.1.3.1 Preparing the Postgres Server ................................................................. 44
        3.1.3.1.1 Installing the sslutils Extension ....................................................... 44
        3.1.3.1.2 Creating a Service Script or Registering the Service ..................... 47
      3.1.3.2 Invoking the PEM Server Installer ......................................................... 49
    3.1.4 Using the PEM Web Client ................................................................................. 62
  3.2 Installing the PEM Client ........................................................................................ 65
  3.3 Installing a PEM Agent ............................................................................................. 71

4 Adding Postgres Instances to PEM .............................................................................. 80
  4.1 Logging on to the PEM server .................................................................................. 80
  4.2 Adding a New Server to the Directory .................................................................... 86
  4.3 Verifying the Connection and Binding .................................................................... 95

5 Installing the SQL Profiler Plugin ............................................................................. 98
  5.1 Installing SQL Profiler ........................................................................................... 99
  5.2 Configuring SQL Profiler ....................................................................................... 104

6 Managing a PEM Installation ..................................................................................... 106
  6.1 Assigning Privileges to PEM Users ......................................................................... 106
  6.2 Managing PEM Authentication .............................................................................. 108

7 Upgrading and Uninstalling PEM Components ......................................................... 110

8 Server Configuration ................................................................................................... 113
  8.1 Modifying the pg_hba.conf File .............................................................................. 113
8.2 Controlling the PEM Server or PEM Agent on Linux.................................115
8.3 Controlling the PEM Server or PEM Agent on Windows............................116
8.4 Controlling the Apache/PHP Server.............................................................117
9 Troubleshooting................................................................................................119
  9.1 Connection to the PEM Server failed..............................................................119
10 Appendix A, Linux Service Script (Sample)......................................................121
1 Introduction

Postgres Enterprise Manager™ (PEM) is designed to assist database administrators, system architects, and performance analysts when administering, monitoring, and tuning PostgreSQL and Postgres Plus 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 Postgres Plus Advanced Server installation, where either is appropriate.
1.1 *Typographical Conventions Used in this Guide*

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

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

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

Copy right © 2012 EnterpriseDB Corporation. All rights reserved.
2 Postgres Enterprise Manager™ - Overview

Postgres Enterprise Manager™ (PEM) is comprised of a number of individual components that must be installed in order to make full use of all the 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 providing web services for the client.

- **PEM Client:** The PEM client is a graphical client application that allows you to manage your Postgres server, and access monitoring data on the PEM server. This program may reside on the PEM server host, or other machines for more convenience.

- **PEM Agent:** A PEM agent runs on each monitored host. PEM agents are responsible for executing tasks and reporting statistics from the host and monitored Postgres instances to the PEM server. A single PEM agent can monitor multiple installed instances of Postgres on each host.

- **SQL Profiler plugin:** This plugin to the Postgres server is used to generate the monitoring data used by the SQL Profiler tool within the PEM client. 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 a Postgres Plus 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, PEM Client and PEM agent are supported on the following platforms:

32 bit Windows

- Windows Server 2008 R1

64 bit Windows

- Windows Server 2008 R1
- Windows Server 2008 R2

32 and 64 bit Linux

- CentOS 5.x and 6.x
- Red Hat Enterprise Linux 5.x and 6.x
- SLES 11.x
- Ubuntu 12.04 LTS

The PEM agent is also supported on the following platforms:

- HP-UX versions 11i v3-1009 and above running on Itanium based hardware (the PEM agent is not supported on the PA-RISC architecture).
- Solaris SPARC 64, v 10.x and 11.x
- Solaris x86-64, v 10.x and 11.x

Our labs and testing groups frequently use the following platforms, but they are not supported in a production environment:

- Windows 7 (32 and 64 bit)
- Windows Vista (32 and 64 bit)
- Windows XP (32 bit)
2.2 Pre-Requisites

SELinux Permissions

Before installing Advanced 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 Postgres Plus Advanced 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
```

Windows Permissions

Depending on your version of 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.

Firewall Restrictions

Please note that you must adjust your firewall to allow communication between PEM components.
3 Installing Postgres Enterprise Manager™

Postgres Enterprise Manager™ installers are available via StackBuilder, or from the EnterpriseDB website at:

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

To install all of the Postgres Enterprise Manager™ components and take full advantage of PEM functionality, you will need three installers, and the SQL Profiler plug-in:

The PEM server installer:

```
pep_server-3.x.x-x-platform.ext
```

The PEM client installer:

```
pep_client-3.x.x-x-platform.ext
```

The PEM agent installer:

```
pep_agent-3.x.x-x-platform.ext
```

The SQL Profiler plug-in:

```
sqlprofiler-pg-9.x.x-x-platform.ext
```

Where:

```
platform
```

(platform is the name of the platform on which you will install the component.

```
ext
```

(ext is the file extension associated with the platform on which you will install the component.)
3.1 Installing 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 server must be version 9.0 or later. The server on which the PEM server resides must 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 for 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 9.2, 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.

Apache/PHP is made available for Postgres installations through the PEM Server installer or the StackBuilder utility. If Apache/PHP 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 Apache/PHP installation, the PEM server installer will add it.

Before installing the PEM server, you must decide if you wish to run PostgreSQL and Apache/PHP on the same host or on separate hosts. If you intend to run the PostgreSQL database server and Apache/PHP 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 PEM web client (shown in Figure 3.1).
You can use the PEM web client to review information about objects that reside on monitored servers, or to review statistical information gathered by the PEM server in a browser.

The PEM web client can help preserve security on monitored servers by providing access to statistical and status information about objects that reside on a monitored server without requiring an actual connection to each server. Please note that the PEM web client displays information gathered by the PEM agents, rather than displaying a current view from system catalogs on the monitored servers.

For more information about using the PEM web client, see Section 3.1.4, Using the PEM Web Client.
3.1.1 Installing the PEM Server and Apache/PHP 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 an Apache/PHP service that reside on the same host. In this configuration, the PEM server installer will provide the pre-requisite software for the backend host (sslutils), and create a service script (on Linux) or register the service (on Windows).

To 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-3.x.x-x-platform.ext
```

The installer displays a Welcome dialog (shown in Figure 3.2).

![Image of the Welcome dialog](image)

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

Click Next to continue to the License Agreement dialog (shown in Figure 3.3).
Carefully review the license agreement before highlighting the appropriate radio button and accepting the agreement; click Next to continue to the Registration required dialog (shown in Figure 3.4).
The Registration required... dialog asks if you have already signed up for an EnterpriseDB user account; select the appropriate option before clicking Next.

![Registration required at www.enterprisedb.com](image)

If you do not have an account, click 'Next' to register at the EnterpriseDB website. Registration will also give you access to many free assets at EnterpriseDB including:

- PostgreSQL Performance Tuning (Bruce Momjian On-Demand Webcasts)
- Top 5 Ways to Supercharge your Database (On-Demand Webcast)
- Data Processing Inside PostgreSQL (Bruce Momjian On-Demand Webcasts)

○ I do not have an enterprisedb.com user account

○ I already have an enterprisedb.com user account

---

**Figure 3.4 - Select a registration option.**

If you have not signed up for an EnterpriseDB user account, clicking Next will open a web browser, displaying the EnterpriseDB User Login/Registration page of the EnterpriseDB website (shown in Figure 3.5).
Figure 3.5 - Register as an EnterpriseDB user.

The registration page can also be accessed directly through:

https://www.enterprisedb.com/user-login-registration

Provide your user information; when you’ve completed the form, return to the EnterpriseDB User Account Information dialog (shown in Figure 3.6).
Figure 3.6 - Enter your EnterpriseDB user account information.

After providing your registered Email address and Password, click Next to continue to the Installation Directory dialog (shown in Figure 3.7).
Figure 3.7 - Specify an installation directory.

Use the Installation Directory dialog (shown in Figure 3.7) to specify the location of the PEM server:

- By default, the PEM server is installed in the /home/opt/PEM directory. 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.

- Use the Show advanced options check box to instruct the installer to open the Advanced options dialog.

The Advanced options dialog is used when installing the PostgreSQL database server and Apache/PHP 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 Apache/PHP 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 the EnterpriseDB distributions of Apache/PHP and PostgreSQL 9.2. If the installer does not locate these packages, it will inform you in the **Dependency missing** dialog (shown in Figure 3.8).

![Dependency missing](image)

**Figure 3.8 - The installer checks for pre-requisites.**

If the installer cannot locate Apache/PHP and/or PostgreSQL 9.2, the PEM server installer will launch the respective installation wizards; follow the onscreen directions presented by the installation wizards for each package.

After installing Apache/PHP and/or PostgreSQL (as needed), the installation process continues by displaying the **PostgreSQL Installation Details** dialog, as shown in Figure 3.9.
The information provided on the PostgreSQL Installation Details dialog enables the installer to connect to the PostgreSQL server.

![Database Server Installation Details](image)

**Figure 3.9 - Enter the name and password of the PostgreSQL superuser.**

Provide the **User** name and **Password** of a PostgreSQL database superuser. After supplying the requested information, click **Next** to continue to the Network Details dialog (as shown in Figure 3.10).
Use the **Network Details** dialog to specify the CIDR-style network address from which the PEM agents will connect to the server (the *client-side* address).

![Network Details dialog](image)

**Figure 3.10 - Supply the network address from which the agent will connect.**

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 (shown in Figure 3.11).
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.

![Agent Details](image)

*Figure 3.11 - 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 (shown in Figure 3.12).
The wizard is now ready to install the PEM server.

Figure 3.12 - The installation is ready to begin.

Click Back to modify any of the options previously selected, or Next to continue with the installation.
During the installation process, the installer will copy files to the system, and set up the database and web services required to run PEM. A popup dialog opens (shown in Figure 3.14) confirming that the \texttt{pem} database has been created and configured.

\textit{Figure 3.13 - The installation in progress.}

Click \texttt{OK} to acknowledge that the \texttt{pem} database has been created, and continue to the \texttt{Completed...} dialog (shown in Figure 3.15).

\textit{Figure 3.14 - Click OK to confirm that the pem database has been created.}
Figure 3.15 - The PEM server installation is complete.
3.1.2 Installing the PEM Server and Apache/PHP on Separate Hosts

To use separate hosts for the PEM server backend database and the Apache/PHP server 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 Apache/PHP server.

3. Invoke the PEM server installer on the host of the Apache/PHP server, selecting the Web Services option on the Installation Type dialog.

To 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-3.x.x-x-platform.ext
```

![Figure 3.16 - The PEM Server Installer's Welcome dialog.](image)
Click Next to continue to the License Agreement dialog (shown in Figure 3.17).

![License Agreement dialog](image)

**Figure 3.17 - The License Agreement.**

Carefully review the license agreement before highlighting the appropriate radio button and accepting the agreement; click Next to continue to the Registration required... dialog (shown in Figure 3.18).
The **Registration required...** dialog (shown in Figure 3.19) asks if you have already signed up for an EnterpriseDB user account; select the appropriate option before clicking **Next**.

![Figure 3.19 - Select a registration option.](image)

If you have not signed up for an EnterpriseDB user account, clicking **Next** will open a web browser, displaying the EnterpriseDB User Login/Registration page of the EnterpriseDB website.

If you have an EnterpriseDB user account, click **Next** to continue to the EnterpriseDB User Account Information dialog (shown in Figure 3.20).
After providing your registered Email address and Password, click Next to continue to the Installation Directory dialog (shown in Figure 3.21).
Figure 3.21 - The Installation Directory dialog.

Use fields on the Installation Directory dialog (shown in Figure 3.21) 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 `/home/opt/PEM` directory. 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 Apache/PHP 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 (shown in Figure 3.22).
Use the radio buttons on the Advanced Options dialog to specify the component or components that you would like to install:

- **Select Web Services and Database** to indicate that the Postgres server and Apache/PHP 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 an Apache/PHP installation.

- **Select Web Services** to install Apache/PHP 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 Apache/PHP.

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 9.0 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 3.23) allows you to specify connection information for the host on which the PEM server backend database (named pem) will reside.

![Advanced options dialog](image)

*Figure 3.23 - The Advanced options dialog.*

Click Next to continue to the Database Server Selection dialog (shown in Figure 3.24).
Use the drop-down listbox on the Database Server Selection dialog 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 9.2 on Port 5432** 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 3.25 opens.

![Database Server Installation Details dialog](image)

**Figure 3.25 - 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 port number on which the Postgres server is listening in the Port field.
- Specify the name of a Postgres database superuser in the User field.
- Specify the password associated with that user in the Password field.
- Specify the name of the service script that controls the specified Postgres server in the Service Name field.

Please note: The PEM server expects a valid service script for the selected server to be located in `/etc/init.d`; for more information about software prerequisites for the PEM server database host, please see Section 3.1.3.1.

Click Next to continue.
Figure 3.26 - Supply the network address from which the agent will connect.

Use the Network Details dialog 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.

You can also specify additional network addresses by manually adding entries to the `pg_hba.conf` file on the PostgreSQL server; for more information, see Section 8.1.

Accept the default (specifying the localhost), or specify a network address range, and click Next to continue to the Agent Details dialog (shown in Figure 3.27).
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.

Figure 3.27 - 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 (shown in Figure 3.28).

The wizard is now ready to install the PEM server.
Figure 3.28 - The installation is ready to begin.

Click Back to modify any of the options previously selected, or Next to continue with the installation (see Figure 3.29).
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 3.30) confirming that the pem database has been created and configured.

Figure 3.30 - 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 3.31).
Figure 3.31 - The PEM server installation is complete.

When the database portion of the PEM server installation is completed, you can invoke the PEM server on another host to install (or upgrade) Apache/PHP.
3.1.2.2 Installing Web Services

Selecting the Web Services radio button on the Advanced options dialog (Figure 3.32) instructs the PEM server installer to either update an existing Apache/PHP installation, or install Apache/PHP on the current host.

![Advanced options dialog](image)

_Figure 3.32 - Selecting the Web Services option on the Advanced options dialog._

Before selecting this option, you must have:

- Completed an installation of the PEM server installer, during which you specified a host for the PEM server.

- Modified the `pg_hba.conf` file on the PEM server database host to allow connections from the Apache/PHP host, and restarted the database server. For more information about modifying the `pg_hba.conf` file, see Section 8.1.

When you select the Web Services option and click Next, the PEM server installer will check the current host for an existing Apache/PHP installation. If the installer does not locate an installation, the installer will invoke the Apache/PHP Setup Wizard (see Figure 3.33).
Figure 3.33 - The Apache/PHP installation wizard.

Follow the onscreen instructions of the Apache/PHP Setup Wizard; when the wizard completes the installation, click Finish to open the Database Server Installation Details dialog (shown in Figure 3.34).
Use the fields on the **Database Server Installation Details** dialog 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. The PEM server installer will complete the PEM server installation, adding only those items that must reside on the host of the Apache/PHP server.

---

**Figure 3.34 - The Database Server Installation Details dialog.**

![Database Server Installation Details dialog](image)
3.1.3 Installing the PEM Server on an Existing Postgres Server

You can use an existing Postgres server (version 9.0 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 9.2 (as bundled with the PEM Server installer)
- Postgres Plus Advanced Server 9.2
- Postgres Plus Advanced Server 9.1 (as installed via an RPM package)

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.

If the Postgres server resides on a Linux system, it must include a service script that the PEM server can use to start, stop, and restart the database server. If the Postgres server resides on a Windows system, the Postgres service must be registered with the Windows service manager.

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:

When the web page opens, select the link for the SSLUtils 3.0.0 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, use `make` to build the `sslutils` module:

1. Set the value of `PATH` so it can locate the `pg_config` program:
   
   ```
   export PATH=$PATH:/opt/PostgresPlus/9.2AS/bin/
   ```

2. Then, use `yum` to install the `sslutil` dependencies:
   
   ```
   yum install openssl-devel
   ```

3. Move into the `sslutils` folder, and enter:
   
   ```
   make USE_PGXS=1
   ```

4. Enter:
   
   ```
   make USE_PGXS=1 install
   ```

**On Windows**

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

- PostgreSQL 9.2 (as distributed with the PEM server installer)
- Postgres Plus Advanced Server 9.2

`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 a Postgres Plus Advanced Server installer, you must use Visual Studio to build `sslutils`. If you are using:

- PostgreSQL 9.0 or 9.1, use Visual Studio 2008
- Advanced Server 9.0 or 9.1, use Visual Studio 2008
- PostgreSQL 9.2, use Visual Studio 2010

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 GETTEXTPATH=path_to_gettext
   SET OPENSSLPATH=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. When the build process completes, the `sslutils` directory will contain the following files:

   ```
   sslutils--1.0.sql
   sslutils--unpackaged--1.0.sql
   sslutils--pemagent.sql.in
   sslutils.dll
   ```

   Copy the `.dll` libraries and `.sql` files to:

   **On PostgreSQL 9.0 or Advanced Server 9.0:**
   ```
   COPY sslutils*.sql "%PGPATH%\share\contrib"
   COPY sslutils.dll "%PGPATH%\lib"
   ```

   **On PostgreSQL 9.1 or Advanced Server 9.1:**
   ```
   COPY sslutils*.sql "%PGPATH%\share\extension"
   ```
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 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 in Appendix A, Linux Service Script (Sample), and information about creating a Postgres service and writing a custom Postgres service script at:


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:

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-3.x.x-x-platform.ext
```

The installer displays a Welcome dialog (shown in Figure 3.35).

![Welcome dialog](image)

*Figure 3.35 - The PEM Server Installer's Welcome dialog.*

Click Next to continue to the License Agreement dialog (shown in Figure 3.36).
Carefully review the license agreement before highlighting the appropriate radio button and accepting the agreement; click Next to continue to the Registration required... dialog (shown in Figure 3.36).

Figure 3.36 - The License Agreement.
The **Registration required**... dialog asks if you have already signed up for an EnterpriseDB user account; select the appropriate option before clicking **Next**.

![Registration required at www.enterprisedb.com](image)

**Figure 3.37 - Select a registration option.**

If you have not signed up for an EnterpriseDB user account, clicking **Next** will open a web browser, displaying the EnterpriseDB User Login/Registration page of the EnterpriseDB website.

If you have an EnterpriseDB user account, click **Next** to continue to the EnterpriseDB User Account Information dialog (shown in Figure 3.38).
Figure 3.38 - Enter your EnterpriseDB user account information.

After providing your registered Email address and Password, click Next to continue to the Installation Directory dialog (shown in Figure 3.39).
Figure 3.39 - The Installation Directory dialog.

Use the Installation Directory dialog (shown in Figure 3.39) to specify the location of the PEM server, and open 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.
Use the radio buttons on the Advanced options dialog (shown in Figure 3.40) to specify an installation type. Select:

- **Web Services and Database** if both the Postgres server and the Apache/PHP 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 an Apache/PHP installation, and upgrade or install Apache/PHP if necessary.

- **Web Services** if only the Apache/PHP server will reside on the current host. See Section 3.1.2 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 9.2 server on which PEM will reside.

After selecting an installation option, click **Next** to continue.
Use the drop-down listbox on the Database Server Selection dialog (Figure 3.41) to select a Postgres 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 Postgres Plus Advanced Server installer.

- Select the PostgreSQL 9.2 (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 (version 9.0 or later) 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 Apache/PHP installation, and upgrade or install the service as needed.

If you selected Database on the Advanced options dialog, the Database Server Installation Details dialog (shown in Figure 3.42) opens.

![Database Server Installation Details dialog](image)

**Figure 3.42 - 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.

![Network Details dialog](image)

**Figure 3.43 - Supply the network address from which the agent will connect.**

Use the Network Details dialog (Figure 3.43) 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. For more information about modifying the `pg_hba.conf` file, see Section 8.1.

When you've added the Network address, click Next to continue to the Agent Details dialog (shown in Figure 3.44).
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.

![Agent Details](image)

*Figure 3.44 - 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 (shown in Figure 3.45).

The wizard is now ready to install the PEM server.
Figure 3.45 - The installation is ready to begin.

Click Back to modify any of the options previously selected, or Next to continue with the installation (see Figure 3.46).
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 3.47) confirming that the \texttt{pem} database has been created and configured.

Click \texttt{OK} to acknowledge that the \texttt{pem} database has been created, and continue to the \texttt{Completed…} dialog (shown in Figure 3.48).
Figure 3.48 - The PEM server installation is complete.

If you are using an Apache/PHP service that resides on a separate host:

- Modify the `pg_hba.conf` file on the Postgres server host. See Section 8.1 for more details.

- Invoke the PEM server installer on the host of the Apache/PHP server. See Section 3.1.2.2 for more information about installing Apache/PHP.
3.1.4 Using the PEM Web Client

The PEM server installer has also installed the PEM web client. You can use the PEM web client to review information about objects that reside on monitored servers, or to review statistical information gathered by the PEM server.

To open the PEM web client, navigate through the Applications menu (on Linux) or Start menu (on Windows) to the Postgres Enterprise Manager sub-menu, and select PEM Web Client.

The Postgres Enterprise Manager Web Login dialog (shown in Figure 3.50).

Use the fields on the dialog to authenticate yourself with the PEM server:

- Provide the name of a Postgres user in the Username field.
- Provide the password associated with the user in the Password field.
- Check the box next to Show system objects? to display the contents, properties and statistics of system databases (such as template0) and system schemas (such as pg_catalog).

Figure 3.50 - The Postgres Enterprise Manager Web Login dialog.
Click the **Login** button to connect to the PEM server.

If you are using a **PEM Trial Licence**, the **Product Registration** dialog (Figure 3.51) will open, and offer you the opportunity to register your copy of PEM.

![Product Registration](image)

**Figure 3.51 - The Product Registration dialog.**

Enter the **Product Key**, or press **Cancel** to connect using available time on the **Trial Licence**. The PEM web client opens (as shown in Figure 3.52).

![PEM Web Client](image)

**Figure 3.52 - The PEM Web Client.**
The PEM web client displays information gathered by the PEM agents, rather than displaying a current view from system catalogs on the monitored servers (like the full-featured PEM client). To access online information about using the PEM web client, select Help Contents from the PEM Help menu (see Figure 3.53).

Figure 3.53 - Accessing online usage information about the PEM web client.
3.2 Installing the PEM Client

The PEM client is a graphical interface that allows you to interact with managed and unmanaged database servers and the PEM server. The PEM client may reside on any Linux or Windows workstation.

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

```
pem_client-3.x.x-x-platform.ext
```

The Setup… page opens, welcoming you to the PEM client installer (as shown in Figure 3.54).

![Figure 3.54 - The PEM Client Installer's Welcome dialog.](image)

Click Next to continue to the License Agreement dialog (shown in figure 3.55).
Figure 3.55 - The PEM Client 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 3.56).
Figure 3.56 - Specify an installation directory for the client.

Accept the default installation directory, or select an alternate installation directory for the PEM client and click Next to continue.
Figure 3.57 - The installer is ready to install the PEM Client.

The installation wizard is now ready to install the PEM client (see Figure 3.57). At this point you can click Back to modify the installation directory, or Next to continue.
Figure 3.58 - The installer displays a progress bar to chart the progress of the installation.

The installer displays a progress bar (see Figure 3.58) while it copies files to the system and configures the menu options required to start the PEM client.
Figure 3.59 - The installer confirms that the client installation is complete.

When the installation is complete, the PEM client is ready to run (see Figure 3.59). The PEM client may be invoked from the Start menu (on Windows) or the Applications menu (on Linux).
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 Postgres servers running on the agent's host. A PEM agent must reside on each monitored virtual or physical machine. A copy of the PEM server must be installed and running on the network on which the PEM agent's host resides.

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-3.x.x-x-platform.ext
```

The Setup... page opens, welcoming you to the PEM Agent installer (as shown in Figure 3.60).

![Figure 3.60 - The PEM Agent Installer's Welcome dialog.](image)

Click Next to continue to the License Agreement (shown in Figure 3.61).
Figure 3.61 - The PEM Client 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 3.62).
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, 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.

Click Next to continue to the PEM Server Installation Details dialog (shown in Figure 3.63).
Figure 3.63 - Provide PEM server installation details

Enter the connection details for the PEM server:

- Specify the name or IP address of the host running the PEM server in the Host field (that is, the host running the database server containing the pem database, not the host running the Apache/PHP web server if two separate hosts are being used).

- Specify the name of the Postgres 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. For information about modifying the pg_hba.conf file, see Section 8.1.
Figure 3.64 - Specify a descriptive name for the PEM agent

The PEM client tree control displays the value entered in the Description field (on the Agent Details dialog, shown in Figure 3.64) 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 (such as Acctg).

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 3.65).

![Figure 3.65 – Provide the certificate path](image)

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 3.66 - The PEM Agent installation is ready to begin

Click Next on the Ready to Install dialog (shown in Figure 3.66) to instruct the installer to copy files to the system and register the agent on the PEM server.
Figure 3.67 - Progress bars mark the installation's progress

The PEM agent installer displays progress bars (shown in Figure 3.67) to mark the PEM agent's installation progress.
Figure 3.68 - The PEM Agent installation is complete

When the installation has completed, 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.
4 Adding Postgres Instances to PEM

After installing a PEM server and agent, you can configure PEM to start monitoring PostgreSQL or Postgres Plus Advanced Server instances on any host on which an agent is installed.

**Note**: The server installer will pre-configure an instance of the agent running on the PEM server to monitor the PEM database server, using the default hostname of the PEM server in the connection parameters. If the default hostname is not known to machines running the PEM client, you may need to update the hostname before connecting to the server.

4.1 Logging on to the PEM server

The PEM client (installed with the PEM client installer) provides a convenient graphical interface that allows you to interact with the PEM server and managed (or unmanaged) Postgres instances. To start the PEM client, navigate through the **Applications menu** (on Linux) or **Start menu** (on Windows) to the **Postgres Enterprise Manager™** menu, and select **Postgres Enterprise Manager v3**. The PEM client opens, as shown in Figure 4.1.

![Figure 4.1 - The PEM client dialog.](image-url)
To open the PEM Server Logon dialog (shown in Figure 4.2), click the Server Logon button at the left-most end of the client toolbar, or navigate through the File menu to PEM Server Logon...

![Figure 4.2 - The PEM Server Logon dialog.](image)

Enter the connection details for the PEM server:

- Enter the host name or IP Address where the Apache/PHP server is running in the Host field.
- Enter the port number on which the Apache server is listening (on the Apache/PHP server host) in the Port field.
- By default, the Apache instance on the PEM server listens for encrypted connections on Port 8443. To disable SSL encryption, uncheck the box next to Use SSL?, and specify that the client should connect on Port 8080.

**Note:** The default certificate on the Apache server is self-signed. The PEM client may request additional user confirmation when connecting with SSL encryption enabled for the first time. If a PEM Server connection error dialog opens, use the options presented by the dialog to test the connection transport, confirming that you would like to accept the certificate. When you accept the certificate, the PEM client will allow the connection to the server.

For more information about SSL encryption and Apache/PHP, please see the Apache documentation at:
Enter the name of the connecting role in the **Username** field.

We recommend that you add additional user accounts to the PEM server for everyday use, rather than using the *postgres* user account. You can create new users with the PEM client tool, or use the `CREATE ROLE` SQL statement; new users must be granted membership in the `pem_admin` role.

For more information about PEM user accounts, see Section 6.1, *Assigning Privileges to PEM Users*.

Enter the password associated with the connecting role in the **Password** field.

Check the box next to **Login at startup?** to instruct the PEM client to open the PEM Server Logon dialog when the client starts.

Use the fields on the **SSH Tunnel** tab (shown in Figure 4.3) to specify connection information for an SSH tunnel connection.

![Image of PEM Server Logon dialog](http://httpd.apache.org/docs/2.2/ssl/ssl_howto.html)

*Figure 4.3 - The SSH Tunnel tab.*

Check the box next to **Use SSH tunneling** to make the fields on the **SSH Tunnel** tab active. Then, provide connection information for the SSH tunnel host:

- Specify the name or IP address of the proxy host (through which client connections will be forwarded) in the **Tunnel host** field.
• Specify the name of a user with connection privileges for the tunnel host in the **Username** field.

• Specify the type of authentication that will be used when connecting to the tunnel host in the **Authentication** field. Select the radio button next to **Password** to specify that the PEM client will provide a password for authentication by the tunnel host. This is the default. Select the radio button next to **Identity file** to specify that the PEM client will provide a valid private key file when connecting.

• If the tunnel host is expecting a private key file for authentication, use the **Identity file** field to specify the location of the key file.

• If the tunnel host is expecting a password, use the **Password/Passphrase** field to specify a word or phrase that will be provided to the tunnel host for authentication.

Click **OK** to connect to the PEM server.

**Tip:** Apache/PHP must be running in order for the PEM client to connect to the PEM server. For detailed information about confirming the status of the Apache/PHP server, see Section 8.4, *Controlling the Apache/PHP Server.*

Once connected, the **Global Overview** monitoring dashboard will be displayed. The example shown in Figure 4.4 shows that two agents are running (one on the PEM host itself, and one that was installed elsewhere), and one server is running (the PEM database server). Expanding the **Agents** node in the tree control displays the registered agents.
Figure 4.4 - The Global Overview dashboard.

Tables at the bottom of the Global Overview dashboard display the status of agents, database servers, and current alerts (shown in Figure 4.5).
Figure 4.5 – The Agent Status, Postgres Server Status, and Alert Status tables.
4.2 Adding a New Server to the Directory

A servers may be configured either locally, or in the PEM Server Directory:

- Local servers typically reside on the same machine as the PEM client, and are considered to be unmanaged.
- Servers residing in the PEM Server Directory are considered to be managed, and are available to all PEM users. Servers in the PEM directory will normally be bound to an agent to enable monitoring (and other PEM features).

The process of configuring a server, either managed or unmanaged, is referred to as adding or registering a server.

Please note: You must ensure the `pg_hba.conf` file of the server to be registered allows connections from the host of the PEM client. For information about modifying the `pg_hba.conf` file, please see Section 8.1.

To open the New Server Registration dialog, click on the Add Server button on the client toolbar, or select Add Server from the File menu.

![Figure 4.6 - The Properties tab of the New Server Registration dialog.](image)

Figure 4.6 - The Properties tab of the New Server Registration dialog.
The Properties Tab

Use fields on the Properties Tab (shown in Figure 4.6) to enter the connection details for the new server.

- Enter a descriptive name for the server in the Name field; the descriptive name will be displayed in the tree control.

- Enter the host name or IP address in the Host field.

- Enter the listener port number of the PostgreSQL or Advanced Server host in the Port field.

- Use the Service field to specify the name of a service configured in the pg_service.conf file.

- Select a maintenance database with the drop-down list box in the Maintenance DB field.

- Specify the Username of the user that the new server will use when connecting to the database; the specified user must have membership in the pem-admin role.

- Enter the password associated with the user in the Password field.

- Check the box next to Store password to instruct the PEM client to store the password for future connections.

- Check the box next to Store on PEM Server to instruct the PEM client to store the definition of the server on the PEM server.

  If the connecting role is not a member of pem_admin (they are connected using the read-only pem_user role), or the user has not logged in to the PEM server, the server definition is stored for the current user only.

- Specify a background color for the server using the Colour selector (optional).

- Select a server group for the new server using the drop-down list box in the Group field. The new server will be displayed in the selected group in the PEM client tree control.

- Use the Team field to specify a PostgreSQL role name. Only PEM users who are members of this role, who created the server initially, or have superuser privileges on the PEM server will see this server when they log on to PEM. If this field is left blank, all PEM users will see the server.
Tip: Do not click OK (yet) if you wish to register the server as a managed server; to register the server as a managed server, you must complete the PEM Agent tab, binding the server to a PEM agent.

![New Server Registration dialog](image)

### Figure 4.7 - The SSL tab of the New Server Registration dialog.

**The SSL Tab**

Use fields on the SSL tab (shown in Figure 4.7) to specify any SSL options required.

- Use the drop-down list box in the SSL field to select an SSL mode:

<table>
<thead>
<tr>
<th>Specify</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>require</td>
<td>To require SSL encryption for transactions between the server and the agent.</td>
</tr>
<tr>
<td>prefer</td>
<td>To use SSL encryption between the server and the agent if SSL encryption is available.</td>
</tr>
<tr>
<td>allow</td>
<td>To allow the connection to use SSL if required by the server.</td>
</tr>
<tr>
<td>disable</td>
<td>To disable SSL encryption between the agent and the server.</td>
</tr>
<tr>
<td>verify-ca</td>
<td>To require SSL encryption, and to require the server to authenticate using a certificate registered by a certificate</td>
</tr>
</tbody>
</table>
To require SSL encryption, and to require the server to authenticate using a certificate registered by a trusted certificate authority.

For more information about using SSL encryption, see Section 31.17 of the Postgres documentation.

- If applicable, use the `Server Root Certificate File` field to access a file browser, and specify the location of the server root certificate.

- If applicable, use the `Server Certificate Revocation List` field to access a file browser, and specify the location of the certificate revocation list.

- If applicable, use the `Client Certificate File` field to access a file browser, and specify the location of the client certificate.

- If applicable, use the `Client Key File` field to access a file browser, and specify the location of the client key file.

For more information about implementing SSL support, see Chapter 31.17 of the Postgres Core Documentation, available at the EnterpriseDB website at:

Figure 4.8 - The SSH Tunnel tab of the New Server Registration dialog.

The SSH Tunnel Tab

Use the fields on the **SSH Tunnel** tab (shown in Figure 4.8) to implement SSH tunneling to the new server; check the box next to **Use SSH tunneling** to enable the fields:

- Specify the name or IP address of the proxy host (through which client connections will be forwarded) in the **Tunnel host** field.

- Specify the name of a user with connection privileges for the tunnel host in the **Username** field.

- Specify the type of authentication that will be used when connecting to the tunnel host in the **Authentication** field.

Select the radio button next to **Password** to specify that the PEM client will provide a password for authentication by the tunnel host. This is the default.

Select the radio button next to **Identity file** to specify that the PEM client will provide a valid private key file when connecting.
- If the tunnel host is expecting a private key file for authentication, use the **Identity file** field to specify the location of the key file.

- If the tunnel host is expecting a password, use the **Password/Passphrase** field to specify a word or phrase that will be provided to the tunnel host for authentication.

![New Server Registration Dialog](image)

*Figure 4.9 - The Advanced tab of the New Server Registration dialog.*

**The Advanced Tab**

Use the fields on the **Advanced** tab (shown in Figure 4.9) to specify connection information for the new server:

- Specify the address of the host in the **Host Address** field.

- Check the box next to **Connect now** to instruct the PEM client to negotiate a connection for the new server after you have completed the **New Server Registration** form and clicked **OK**.
• Check the box next to Restore env? to instruct PEM to restore the browser environment when you reconnect to this server. If you regularly use different databases on the same server you may want to deselect this option.

• If you specify a role name in the Rolename field, PEM will connect to the server using the identity specified on the Properties tab; after establishing a connection with the server, the client will assume the identity and permissions of the role specified in the Rolename field. The role specified on the Properties tab must be a member of the role specified on the Advanced tab.

• Specify an SQL restriction in the DB restriction field to limit the databases displayed in the PEM client. For example, enter: 'live_db', 'test_db' to display only live_db and test_db in the PEM browser.

• Specify the name of the PostgreSQL or Advanced Server service in the Service ID field on the Advanced tab. This allows the PEM server to stop and start the service.

  On Unix systems, provide the name of the service script located in /etc/init.d

  On Windows, provide the ID of the service. You can find the service ID in the Services Microsoft Management Console snapin.

The Postgres Plus Advanced Server installer uses a default service ID of ppas-9.x, where x specifies the version number of the server. For example:

  ppas-9.2

The EnterpriseDB one-click installer (for PostgreSQL) uses a default service ID of postgresql-9.x where x specifies the version number of the server. For example:

  postgresql-9.2
The PEM Agent Tab

Use fields on the PEM Agent tab (shown in Figure 4.10) to create a binding between the new server and a PEM agent. A binding provides the link between the PEM server and the PEM agent that allows the PEM server to monitor and manage that agent. The binding also provides the connection parameters that the agent uses when connecting to the PEM server.

It is important to note that the agent may use different connection credentials when connecting to the monitored server than the client uses. For example, the agent on a Unix host may use a Unix Domain Socket to connect to the server, while the client may connect via a connection pooler on an entirely separate host.

- Select an Enterprise Manager agent using the drop-down list box to the right of the Bound Agent label. Each server should be assigned a unique agent.

- Enter the IP address that the agent should use when connecting to the Postgres Plus host in the Host field.

Please Note: You may wish to specify a different Host address on the PEM agent.
tab than the address used on the Properties tab. For example, you can configure the PEM agent to connect via a Unix socket to the monitored server, while the client connects using an SSL connection to a forwarded port on an external firewall (that is not appropriate for the agent).

- Enter the port that the server is listening for connections on. By default, the agent will use the port defined on the Properties tab.

- Use the SSL field to specify an SSL operational mode; specify require, prefer, allow, disable, verify-ca or verify-full.

For more information about using SSL encryption, see Section 31.17 of the Postgres documentation.

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

- Specify the name of the role that agent should use when connecting to the server in the Username field.

- Specify the password that the agent should use when connecting to the server in the Password field.

- Verify the password by re-entering it in the Confirm Password field. If you do not specify a password, you must configure the authentication for the agent manually (for example, by using a .pgpass file).

- Use the Allow takeover option to specify whether or not the monitoring of the server may be taken over by another agent in a High Availability environment.

The PEM client will connect directly to the managed server, so the pg_hba.conf file must contain the appropriate entries to allow those connections. For information about modifying the pg_hba.conf file, see Section 8.1.

When you have specified the server details, click OK to add the connection.
4.3 Verifying the Connection and Binding

If you checked the Connect Now option when defining a new server, the PEM client will initiate a connection to the server when you click OK. You should be able to browse databases and their contents using the tree control on the PEM client.

![Global Overview dashboard](image)

*Figure 4.11 - The Global Overview dashboard.*

When initially connecting to a newly bound server, the Global Overview dashboard may display the new server with a status of “unknown” in the server list (as shown in Figure 4.11) before recognizing the server, the bound agent must execute a number of probes to examine the server, which may take a few minutes to complete.

If, after five minutes, the Global Overview dashboard still does not list the new server, you should review the log for the agent, checking for errors. Right-click the agent in the tree control, and select the Probe Log Analysis option from the Dashboards sub-menu of the context menu.
Figure 4.12 - The Probe Log Analysis dashboard

In the example shown in Figure 4.12, you can see that that agent is unable to connect to the server to which it is bound to due to an authentication issue. You can resolve the problem by modifying the `pg_hba.conf` file to allow server authentication.

Within a few minutes, bar graphs on the Global Overview dashboard shows that the agent has now connected successfully, and the new server is included in the Postgres Server Status list (as shown in Figure 4.13).
Figure 4.13 - The new server can now be used with PEM.
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; since it is a plugin to the database server, you must install a copy of the plugin for each PostgreSQL instance you wish to profile on each host you wish to profile. For example, if you have a host running PostgreSQL 8.4 and PostgreSQL 9.0, 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 8.4 (and above) and Postgres Plus Advanced Server version 9.0 (and above). The plugin is distributed via StackBuilder, or is available from the EnterpriseDB website at:

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

The plugin is also distributed and installed with the server component of the Postgres Plus Advanced Server 9.0 (and above) 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 Postgres Plus Advanced Server 9.0 or above, skip ahead to the Configuration section.
5.1 Installing SQL Profiler

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

```
sqlprofiler-pg-x.x-3.0.x-platform.ext
```

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

![Setup Wizard](image)

*Figure 5.1 - The SQL Profiler Installer Welcome dialog.*

Click Next to continue to the License Agreement, shown in Figure 5.2.
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).
Specify an alternate location for the installation directory, or accept the default location and click Next to continue.

Figure 5.3 - The PostgreSQL Installation Directory
The wizard is now ready to install the SQL Profiler plugin (as shown in Figure 5.4).

![Figure 5.4 - Ready to Install](image)

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

### 5.2 Configuring SQL Profiler

By default, the SQL Profiler plugin is not enabled once installed. This avoids the need to restart the server at the time the initial installation is performed, and prevents the plugin 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, 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 in the PEM client or the psql command line interface, run the sql-profiler.sql script in the database specified as the Maintenance Database in the PEM client. In PostgreSQL this is typically the postgres database; in Postgres Plus Advanced Server edb is the default maintenance database.

The following command invokes the sql-profiler.sql script using the psql command line tool on Linux systems:

```
$ /opt/PostgreSQL/9.2/bin/psql -U postgres postgres < /opt/PostgreSQL/9.2/share/postgresql/contrib/sql-profiler.sql
```

(On Windows systems, this script is located in the share\contrib subdirectory of the main PostgreSQL installation directory.)

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

To open SQL Profiler on the PEM client, highlight the name of a server you wish to profile in the tree control, and select SQL Profiler from the Management menu. The SQL Profiler wizard (shown in Figure 5.7), opens.

![SQL Profiler wizard](image.png)

*Figure 5.7 - The SQL Profiler wizard.*

The SQL Profiler wizard will walk you through the process of defining a new trace, or opening existing trace. For more information about using SQL Profiler, consult the online help text for SQL Profiler (accessed through the PEM client Help menu).
6 Managing a PEM Installation

The PEM client provides a secure environment that you can use to manage Postgres roles. This section details the privileges of those users granted membership in the pem_user role.

For more information about privileges and authentication methods, please consult the Postgres core documentation, available from EnterpriseDB at:


6.1 Assigning Privileges to PEM Users

Users connect to PEM using role credentials created on the PostgreSQL database used by the PEM server. By default, the postgres superuser account is used for the initial logon, however it is strongly recommended that individual roles are created for each connecting user.

You can create a role with the CREATE ROLE SQL command in a command line client, or with the PEM client New Login Role dialog. To access the New Login Role dialog, connect to the PEM server database using the PEM client, and right-click on the Login Roles node and select New Login Role… from the sub-menu (as shown in Figure 6.1).

Figure 6.1 - The context menu of the Login Roles node
Complete the tabs on the New Login Role dialog to add a new PEM user.

![New Login Role dialog](image)

**Figure 6.2 - The Role membership tab of the New Login Role... dialog.**

Use the fields on the Role membership tab (shown in Figure 6.2) to grant new roles appropriate permissions and role memberships to use PEM:

- Users that are members of the `pem_user` role may view dashboards and use tools like the Postgres Expert and Capacity Manager, but they will not be able to install agents or configure the server, directory, alerts or probes.

- Users that are members of the `pem_admin` role have the same read permissions as members of the `pem_user` role, plus sufficient privileges to configure the server, directory, alerts and probes.

- Administrative users must be added to the `pem_admin` role and explicitly granted the `CREATE ROLE` privilege. In addition to the permissions granted through
membership in the `pem_admin` role, the `CREATE ROLE` privilege allows an administrator to create additional PEM users, and to install and register new agents.

For detailed information about using the tabs on the `New Login Role` dialog, click the `Help` button in the lower left corner of the dialog.

For more information about using the `CREATE ROLE` command, see the PostgreSQL core documentation at:


### 6.2 Managing PEM Authentication

Postgres supports a number of authentication methods:

- Secure password (md5)
- GSSAPI
- SSPI
- Kerberos
- Ident
- LDAP
- RADIUS
- Certificate (SSL)
- PAM

Postgres (and PEM) authentication is controlled by the `pg_hba.conf` configuration file. Entries within the configuration file specify who may connect to a specific database, and the type of authentication required before that user is allowed to connect.

A typical entry in the `pg_hba.conf` file that allows a user named `postgres` to connect to all databases from the local host (127.0.0.1/32) using secure password (md5) authentication connections would take the form:

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

For more information about modifying the `pg_hba.conf` file, see [Section 8.1](#).

Depending on your system's configuration, you may also need to create a password file for the user account that the PEM agent uses to connect to the server, to allow the agent to properly respond to the server's authentication request. An entry in the password file for a user named `postgres`, with a password of `1safepwd` could take the form:

```
localhost:5432:*:postgres:1safepwd
```
The password file is usually named `~root/.pgpass` on Linux systems, or `%APPDATA%\postgresql\pgpass.conf` (on Windows). For more information about configuring a password file, visit the EnterpriseDB website at:


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

7 Upgrading and Uninstalling PEM Components

PEM documentation and links to PEM and SQL Profiler installers are available from the EnterpriseDB website at:


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.
3. Invoke the PEM client installer on any server that is hosting a PEM client.

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.

For information about controlling the server, please see Section 8.3, Controlling a Service on Windows.

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 from a PEM client.

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:

<table>
<thead>
<tr>
<th>Component</th>
<th>Uninstaller name</th>
<th>Default location</th>
</tr>
</thead>
<tbody>
<tr>
<td>PEM server</td>
<td>uninstall-pemserver</td>
<td>/opt/PEM/server</td>
</tr>
<tr>
<td>PEM client</td>
<td>uninstall-pemclient</td>
<td>/opt/PEM/client</td>
</tr>
<tr>
<td>PEM agent</td>
<td>uninstall-pemagent</td>
<td>/opt/PEM/agent</td>
</tr>
<tr>
<td>SQL Profiler</td>
<td>uninstall-sqlprofiler</td>
<td>/opt/PostgreSQL/9.2</td>
</tr>
</tbody>
</table>

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 Add/Remove Programs application to remove PEM components. Select the Add/Remove Programs option from the Windows Control Panel; when the control panel opens, locate the name of the PEM component in the program list. Click the Remove button to remove the component.
8 Server Configuration

Changes to the `pg_hba.conf` authentication configuration file require a server reload before the change will take effect. The sections that follow provide information about modifying the `pg_hba.conf` file, restarting the PEM server, and controlling the Apache/PHP service on Linux and Windows.

8.1 Modifying the `pg_hba.conf` File

Entries in the `pg_hba.conf` file control network authentication and authorization. The `pg_hba.conf` file on the PEM server host must allow connections between the PEM server and Apache/PHP, the PEM agent, and the monitored servers.

During the PEM server installation process, you are prompted for the IP address and connection information for hosts that will be monitored by PEM; this information is added to the top of the `pg_hba.conf` file (see Figure 8.1).

```
# Allow remote PEM agents and users to connect to PEM server
hostssl pem +pem user 192.168.2.8/24 md5
hostssl pem +pem agent 192.168.2.8/24 cert
# Allow local PEM agents and admins to connect to PEM server
hostssl pem +pem user 127.0.0.1/32 md5
hostssl postgres +pem user 127.0.0.1/32 md5
hostssl pem +pem user 127.0.0.1/32 md5
hostssl pem +pem agent 127.0.0.1/32 cert
# PostgreSQL Client Authentication Configuration File
# -----------------------------------------------
# # Refer to the "Client Authentication" section in the PostgreSQL
# # documentation for a complete description of this file. A short
# # synopsis follows.
# # This file controls, which hosts are allowed to connect, how clients
# # are authenticated, which PostgreSQL user names they can use, which
# # databases they can access. Records take one of these forms:
# # local DATABASE USER METHOD [OPTIONS]
# host DATABASE USER ADDRESS METHOD [OPTIONS]
# hostssl DATABASE USER ADDRESS METHOD [OPTIONS]
```

Figure 8.1 - PEM entries in the `pg_hba.conf` file.

You may also need to manually modify the `pg_hba.conf` file to allow connections between the PEM server and other PEM components. For example, if your Apache/PHP installation does not reside on the same host as the PEM server, you must modify the `pg_hba.conf` file on the PEM server host to allow Apache/PHP to connect to the server.
By default, the `pg_hba.conf` file resides in the data directory, under your PostgreSQL installation; for example, on a PostgreSQL 9.2 host, the default location of the `pg_hba.conf` is:

```
/opt/PostgreSQL/9.2/data
```

You can modify the `pg_hba.conf` file with your editor of choice. After modifying the file, restart the server for changes to take effect. For information about restarting the server, see Section 8.2 (for Linux) or Section 8.3 (for Windows).

The following example shows a `pg_hba.conf` entry that allows an md5 password authenticated connection from a user named `postgres`, to the `postgres` database on the host on which the `pg_hba.conf` file resides. The connection is coming from an IP address of 192.168.10.102:

```
# TYPE  DATABASE        USER            CIDR-ADDRESS            METHOD
# IPv4 local connections:
host    postgres        postgres        192.168.10.102/32       md5
```

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

For more information about formatting a `pg_hba.conf` file entry, please see the PostgreSQL core documentation at:

```
```
8.2 Controlling the PEM Server or PEM Agent on Linux

On Linux platforms, service scripts reside in the `/etc/init.d` directory. The default name of the service script that controls:

- A PEM server host on Advanced Server is `ppas-9.x`.
- A PEM server host on PostgreSQL is `postgresql-9.x`.
- A PEM agent is `pemagent`.

To determine if a service is running, open a command line, and issue the command:

```
/etc/init.d/service_name status
```

Where `service_name` is the name of the service.

If prompted, enter a password; if the service is running, Linux will respond:

```
pg_ctl: server is running (PID: 13957)
/home/opt/PostgreSQL/9.2/bin/postgres "-D"
"/home/opt/PostgreSQL/9.2/data"
```

If the service is not running, Linux will respond:

```
pg_ctl: no server running
```

You can use the service script to start the service. Enter:

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

Where:

- `service_name` `service_name` is the name of the service.
- `action` specifies the action taken by the service command. Specify:
  - `start` to start the service.
  - `stop` to stop the service.
8.3 Controlling the PEM Server or PEM Agent on Windows

The Windows operating system includes a graphical service controller that displays the server status, and offers point-and-click server control. The registered name of the service that controls:

- A PEM server host on PostgreSQL is `postgresql-9.x`
- A PEM server host on Advanced Server is `ppas-9.x`
- A PEM agent is `pemagent`

Where `x` indicates the server version number.

The Services utility can be accessed through the Windows Control Panel. When the utility opens, use the scroll bar to navigate through the listed services to highlight the service name (see Figure 8.2).

![Services Utility](image.jpg)

Figure 8.2 – The Advanced Server service in the Windows Services window.

The *Stop the service* option stops the instance of Advanced Server. Any user (or client application) connected to the Postgres server will be abruptly disconnected if you stop the service.

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

**Use the Start the service option to start the Advanced Server service.**

### 8.4 Controlling the Apache/PHP Server

On Linux, you can confirm the status of the Apache/PHP service by opening a command line, and entering the following command:

```
ps -ef | grep apache
```

If Linux responds with an answer that is similar to the following example, Apache/PHP is not running:

```
user     13321 13267  0 16:37 pts/1    00:00:00 grep apache
```

You can use the following command to start Apache/PHP:

```
/etc/init.d/EnterpriseDBApachePhp start
```

If Apache/PHP starts properly, the `ps -ef` command will result in output similar to the following:

```
$ ps -ef | grep apache
root     13712     1  0 16:40 ?        00:00:00 /opt/PostgreSQL/EnterpriseDB-
ApachePhp/apache/bin/httpd -k start -f /opt/PostgreSQL/EnterpriseDB-
ApachePhp/apache/conf/httpd.conf
daemon   13744 13712  0 16:40 ?        00:00:00 /opt/PostgreSQL/EnterpriseDB-
ApachePhp/apache/bin/httpd -k start -f /opt/PostgreSQL/EnterpriseDB-
ApachePhp/apache/conf/httpd.conf
daemon   13745 13712  0 16:40 ?        00:00:00 /opt/PostgreSQL/EnterpriseDB-
ApachePhp/apache/bin/httpd -k start -f /opt/PostgreSQL/EnterpriseDB-
ApachePhp/apache/conf/httpd.conf
daemon   13746 13712  0 16:40 ?        00:00:00 /opt/PostgreSQL/EnterpriseDB-
ApachePhp/apache/bin/httpd -k start -f /opt/PostgreSQL/EnterpriseDB-
ApachePhp/apache/conf/httpd.conf
daemon   13747 13712  0 16:40 ?        00:00:00 /opt/PostgreSQL/EnterpriseDB-
ApachePhp/apache/bin/httpd -k start -f /opt/PostgreSQL/EnterpriseDB-
ApachePhp/apache/conf/httpd.conf
daemon   13748 13712  0 16:40 ?        00:00:00 /opt/PostgreSQL/EnterpriseDB-
ApachePhp/apache/bin/httpd -k start -f /opt/PostgreSQL/EnterpriseDB-
ApachePhp/apache/conf/httpd.conf
user     13790 13267  0 16:41 pts/1    00:00:00 grep apache
```

On Windows, you can use the `Services` application to check the status of the Apache/PHP service.

To confirm the status of the Apache/PHP service, navigate through the **Control Panel,** to **System and Security,** select **Administrative Tools,** and then **Services.**
The Apache/PHP server runs as a service named EnterpriseDB ApachePHP (see Figure 8.3).

![Figure 8.3 - The EnterpriseDB ApachePHP Windows service.](image)

The **Status** column displays the current state of the Apache/PHP server. Click the **Start** link to start Apache/PHP if the service is not running.
9 Troubleshooting

9.1 Connection to the PEM Server failed

When connecting to the PEM server with the full-featured PEM client for the first time with SSL encryption enabled, you may encounter the error message shown in Figure 9.1.

![Figure 9.1 - Connection to the PEM Server failed.](image1)

Click Yes (see Figure 9.2) to test the connection to the server.

![Figure 9.2 - The PEM server connection test.](image2)

The test prompts you to confirm the validity of any unknown or expired certificates (see Figure 9.3).

![Figure 9.3 - The test has detected a self-signed certificate.](image3)
The default certificate on the Apache server is self-signed; click Yes to accept the certificate and continue. The PEM client will confirm that the PEM server connection is working as expected (see Figure 9.4).

![Image of Postgres Enterprise Manager](image)

**Figure 9.4 - The connection test is successful.**

After accepting the SSL certificate, you can select PEM Server Logon from the File menu or click the PEM Server Logon icon to provide credentials to connect to the PEM server.
Appendix A, Linux Service Script (Sample)

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.

```bash
#! /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 9.2"
DB_INSTALL_DIR=/opt/PostgreSQL/9.2
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 "${DB_PID_FILE}" ]; then
    PIDALIVEDB=`ps -p "${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/pv_ctl" status -D "$DB_DATA_DIR" -1 "$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