



EDB Postgres™ Enterprise Manager Upgrade and Migration Guide Version 7.7

March 27, 2019

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

EnterpriseDB Corporation, 34 Crosby Drive, Suite 100, Bedford, MA 01730, USA
T +1 781 357 3390 **F** +1 978 589 5701 **E** info@enterprisedb.com **www**.enterprisedb.com

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

Table of Contents

1	Introduction.....	4
1.1	Typographical Conventions Used in this Guide.....	5
2	Upgrading a PEM Installation.....	6
2.1	Upgrading a PEM Agent.....	7
2.1.1	Using an RPM Package to Upgrade the PEM Agent.....	13
2.2	Upgrading the PEM Server.....	14
2.1.1	Using an RPM Package to Upgrade the PEM Server.....	22
2.3	Upgrading SQL Profiler.....	24
3	Upgrading the Backing Postgres Database.....	25
4	Moving the Postgres Enterprise Manager™Server.....	36
5	Upgrading a Graphical Installation with an RPM Package on a Linux Host.....	49
5.1	Upgrading a PEM Server that was Installed with a Graphical Installer.....	49
5.2	Upgrading a PEM agent Installed with a Graphical Installer.....	52
6	Troubleshooting.....	55
6.1	The pem.alert Table Fails to Restore.....	55
7	Uninstalling Postgres Enterprise Manager™.....	56

1 Introduction

This document provides detailed information about upgrading the Postgres Enterprise Manager™ (PEM) server:

- **Upgrading a PEM Installation** - Section [2](#) provides information about upgrading your PEM server from one major version to another (i.e. from 6.0 to 7.5).
- **Upgrading the Backing Database** - Section [3](#) provides detailed information about upgrading the backing database, while maintaining the same version of the PEM Server.
- **Moving a PEM Server** –Section [4](#) provides detailed information about moving the PEM server from one host to another host.

Please note that in future PEM releases, a graphical installer will not be available for the PEM agent or server on a Linux host; you will be required to use an RPM package to install or update a PEM installation. For detailed information about using an RPM package to update the Linux host of a PEM server or agent that was installed with the graphical installer, please see Section [5](#).

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

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 preceding term may be repeated. For example, [a | b] ... means that you may have the sequence, “b a a b a”.

2 Upgrading a PEM Installation

You can use PEM graphical installers to update a PEM agent or server that was installed with a graphical installer to a more-recent version. Links to PEM and SQL Profiler installers are available at the EnterpriseDB website at:

<http://www.enterprisedb.com/products-services-training/products/postgres-enterprise-manager>

Upgrading Postgres Enterprise Manager™ components

To upgrade PEM component software, simply invoke a newer version of the PEM component installers in the following order:

1. Invoke the PEM agent installer on each monitored node *except* the PEM server node. Please note that your upgraded PEM 7.5 agents will work with the PEM 6 server while the upgrade is being performed.
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.

During an installation, the component installer will automatically detect an existing installation, and perform an upgrade. After upgrading the PEM components, you should upgrade SQL Profiler; this step is platform-specific.

As of PEM 7, the PEM client interface is installed as part of the PEM server component, and is accessed with your choice of web browser. For detailed information about using the PEM 7 web interface, please see the *PEM Getting Started Guide*.

Please note that if you have used an RPM package to install your server or agent, you must use an RPM package to upgrade the installation. Likewise, if you initially installed with a graphical installer, you must upgrade with a graphical installer.

The following sections will walk you through the upgrade process, step-by-step.

2.1 Upgrading a PEM Agent

To upgrade a system that is currently monitored by a PEM agent to a more-recent version of PEM agent, simply download and invoke a newer version of the platform-specific PEM agent installer on the system that the agent is monitoring.

If you are a Linux user, you can invoke the installer by opening a terminal window, assuming superuser privileges, navigating to the directory in which the installer resides and entering:

```
./pem_agent-x.x.x-linux.run
```

Where *x.x.x* specifies the version information for the installer.

If you are a Windows user, you can invoke the installer by right-clicking on the downloaded installer's icon, and selecting *Run as Administrator*.



Figure 2.1 - The PEM Agent installer welcome window.

The PEM Agent Setup Wizard opens, welcoming you (see Figure 2.1).



Figure 2.2 - The PEM license agreement.

Read and accept the License Agreement (shown in Figure 2.2) before clicking Next to continue.

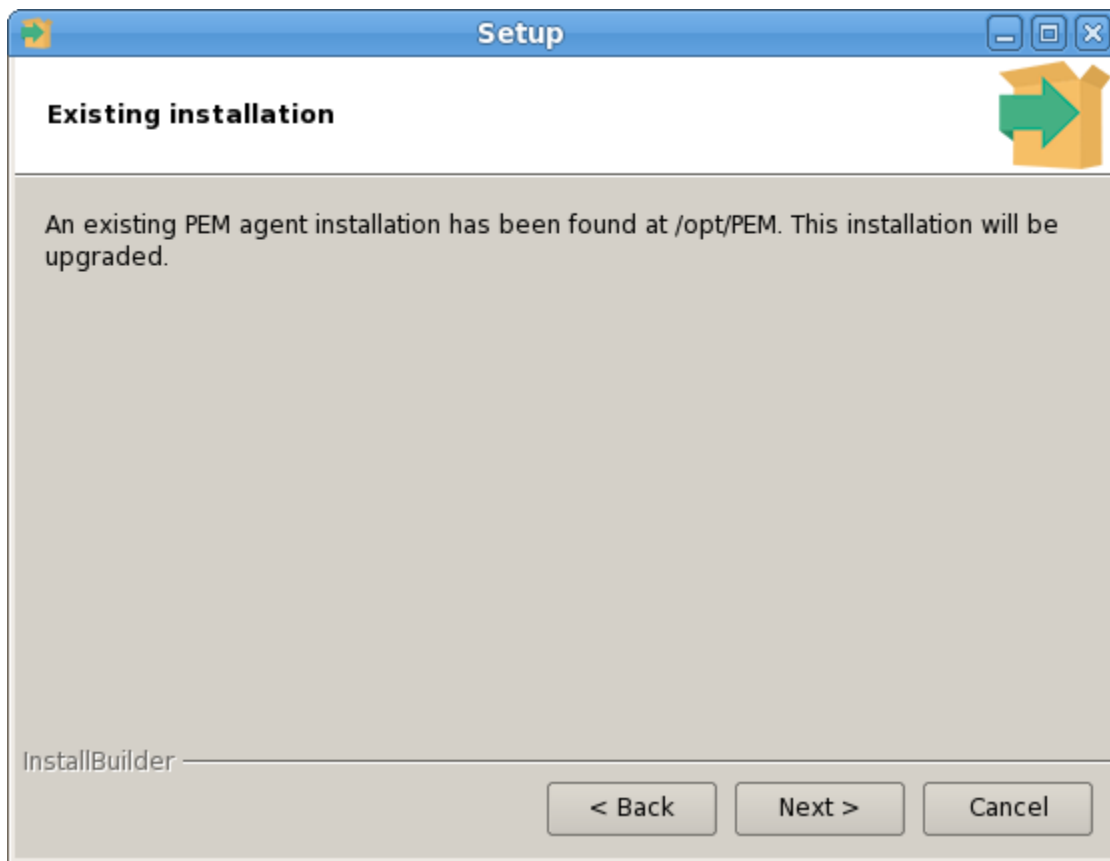


Figure 2.3 - The installer detects an existing installation.

The setup wizard will automatically detect an existing agent, and upgrade the installed version (see Figure 2.3). Click `Next` to continue.

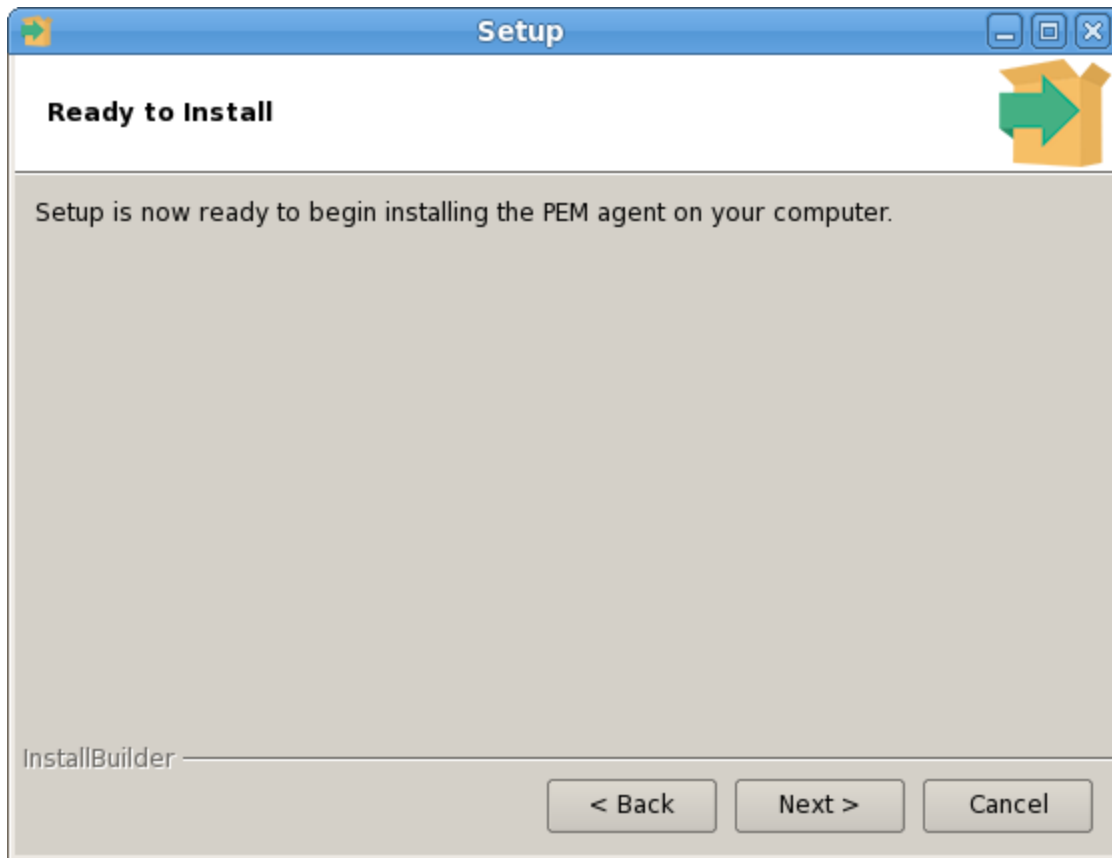


Figure 2.4 - The PEM Agent will perform the upgrade.

When the `Ready to Install` dialog (shown in Figure 2.4) informs you that the installation is about to begin, click `Next` to continue.

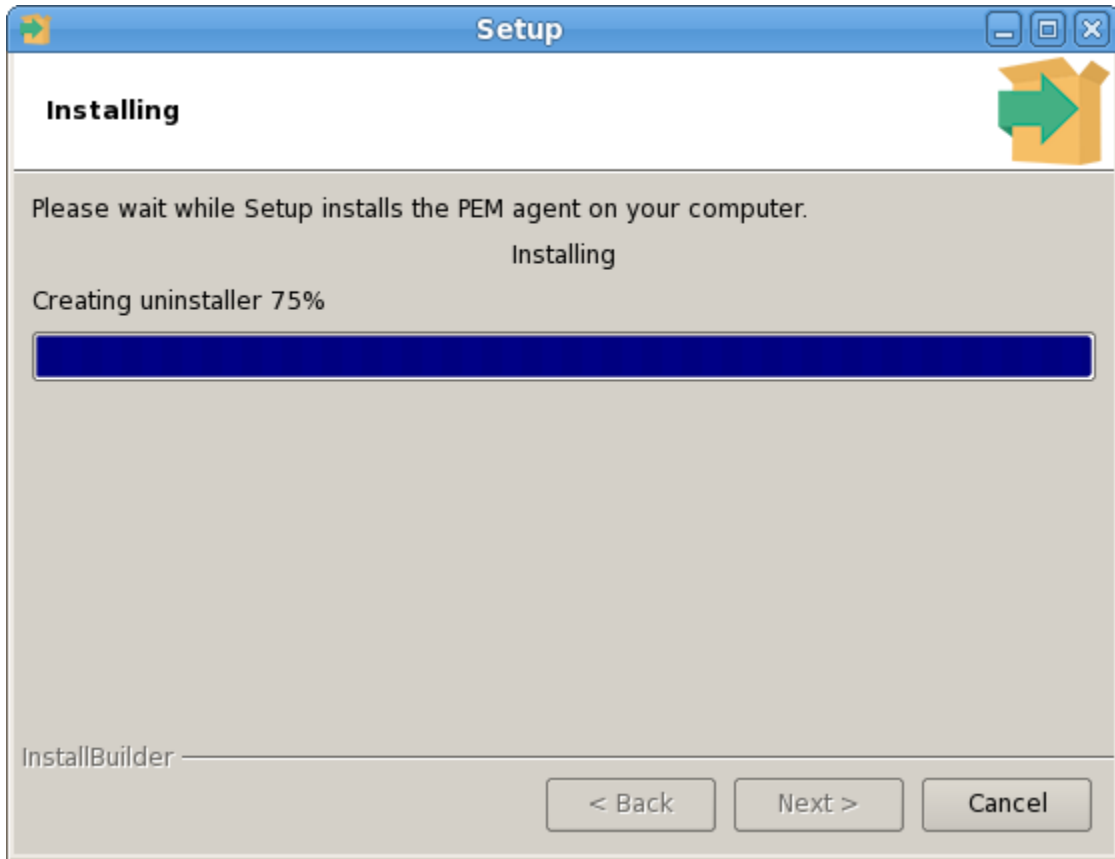


Figure 2.5 - Progress bars chart the installation process.

The setup wizard displays progress bars to inform you of each component that is being installed (see Figure 2.5).

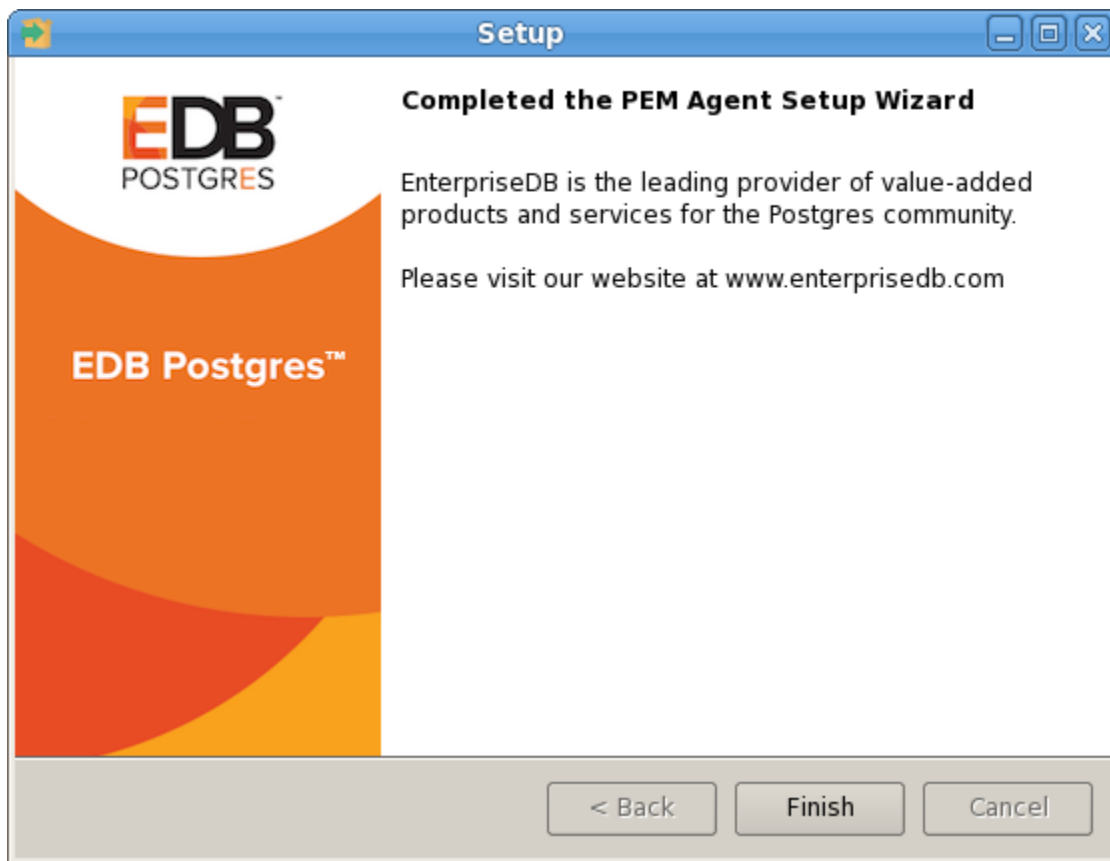


Figure 2.6 - The PEM Agent Setup Wizard has finished the update.

The PEM Agent Setup Wizard will inform you when the installation completes (see Figure 2.6). Click **Finish** to exit the wizard and close the window.

2.1.1 Using an RPM Package to Upgrade the PEM Agent

You can use an RPM package to upgrade existing agents that were initially installed by a package. The upgrade process does not update the PEM agent configuration file. After installing the new agent, you must manually copy the configuration file of the existing agent to the new installation location.

To use RPM packages to upgrade a PEM agent, you must:

1. Upgrade the installed version of the PEM agent:

```
yum upgrade pem-agent
```

2. If upgrading from PEM 6.x, copy the configuration file to the location of the PEM 7.x configuration file (`/usr/pem/agent/etc/agent.cfg`). For example:

```
cp /usr/pem-6.0/etc/agent.cfg /usr/pem/agent/etc/agent.cfg
```

For detailed information about using an RPM package to install the PEM agent, please see the *PEM Installation Guide* or the *Advanced Server Installation Guide*, available at:

<http://www.enterprisedb.com/products-services-training/products/documentation/enterpriseedition>

2.2 Upgrading the PEM Server

The PEM server installer facilitates upgrading directly between major versions of the PEM server (for example, you can upgrade directly from version 5.0 to version 7.5 without first upgrading to version 6.0). During the server installation, the setup wizard also performs a dependency check to ensure that the Apache/PHP version is upgraded to the version required by the new PEM server version.

If you are a Linux user, you can invoke the downloaded PEM server installer by opening a terminal window, assuming superuser privileges, navigating to the directory in which the installer resides and entering:

```
./pem_server-x.x.x-linux.run
```

Where *x.x.x* specifies the version information for the installer.

If you are a Windows user, you can invoke the installer by right-clicking on the downloaded installer's icon, and selecting `Run as Administrator`.



Figure 2.7 - The PEM Server Setup Wizard.

The PEM Server Setup Wizard welcomes you, as shown in Figure 2.7. Click **Next** to continue to the License Agreement.

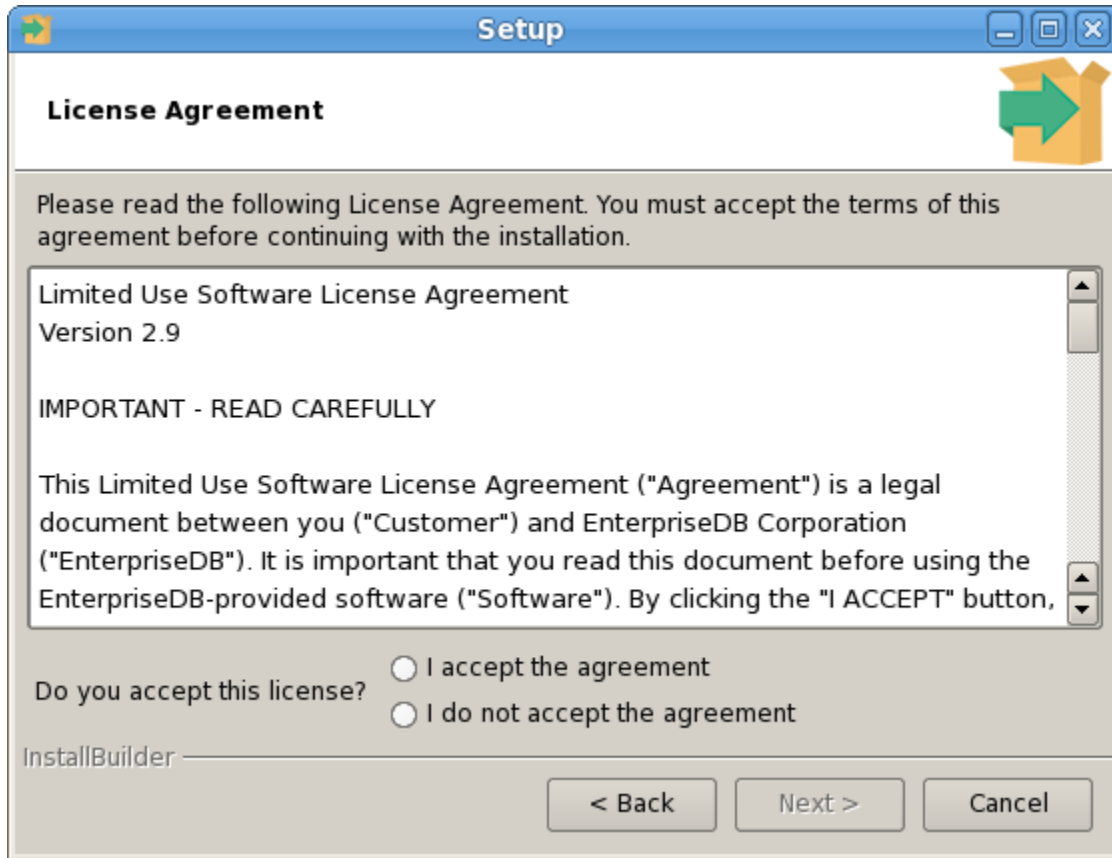


Figure 2.8 - Accept the License Agreement to continue.

The PEM server setup wizard will prompt you to accept the License Agreement (shown in Figure 2.8). After reviewing the license agreement, check the radio button next to **I accept the agreement**, and click **Next** to continue to the Existing installation dialog.

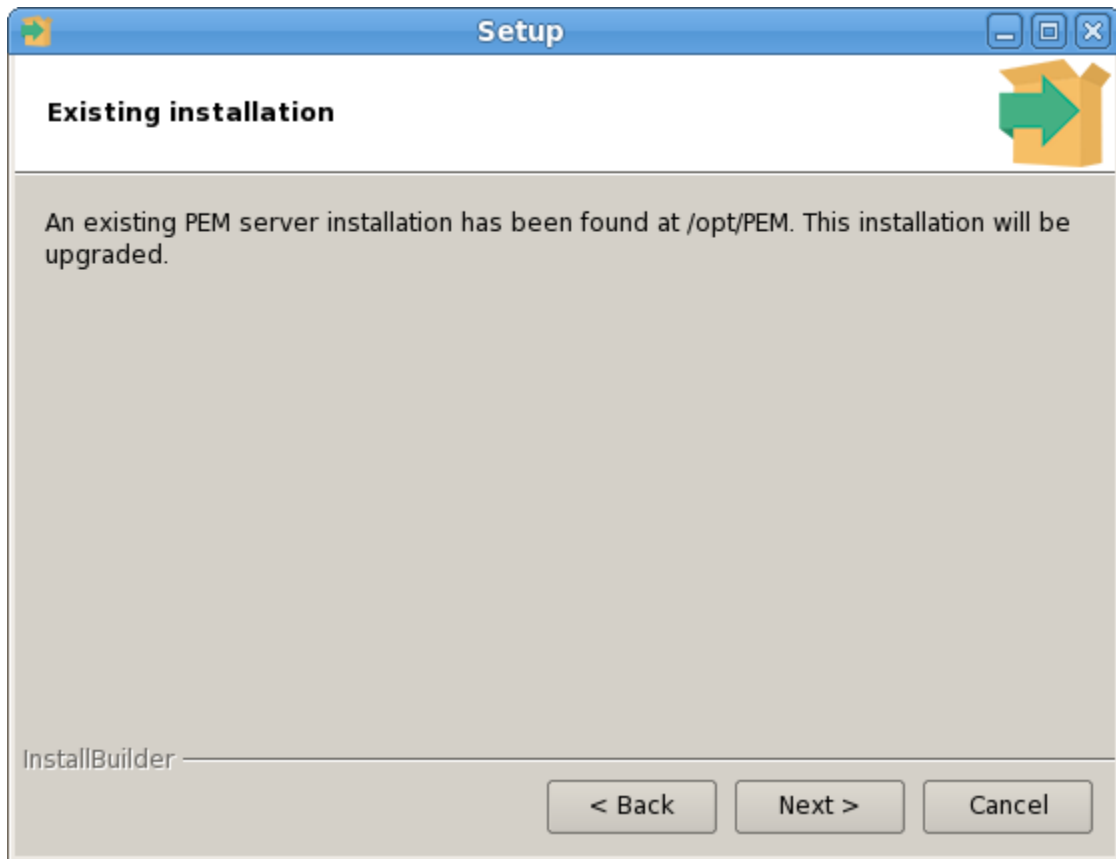


Figure 2.9 - The PEM server installer detects an existing PEM server installation.

The wizard will check the PEM server host for an existing PEM server installation; if the wizard locates an installation, it will perform an upgrade (see Figure 2.9). Click **Next** to continue.

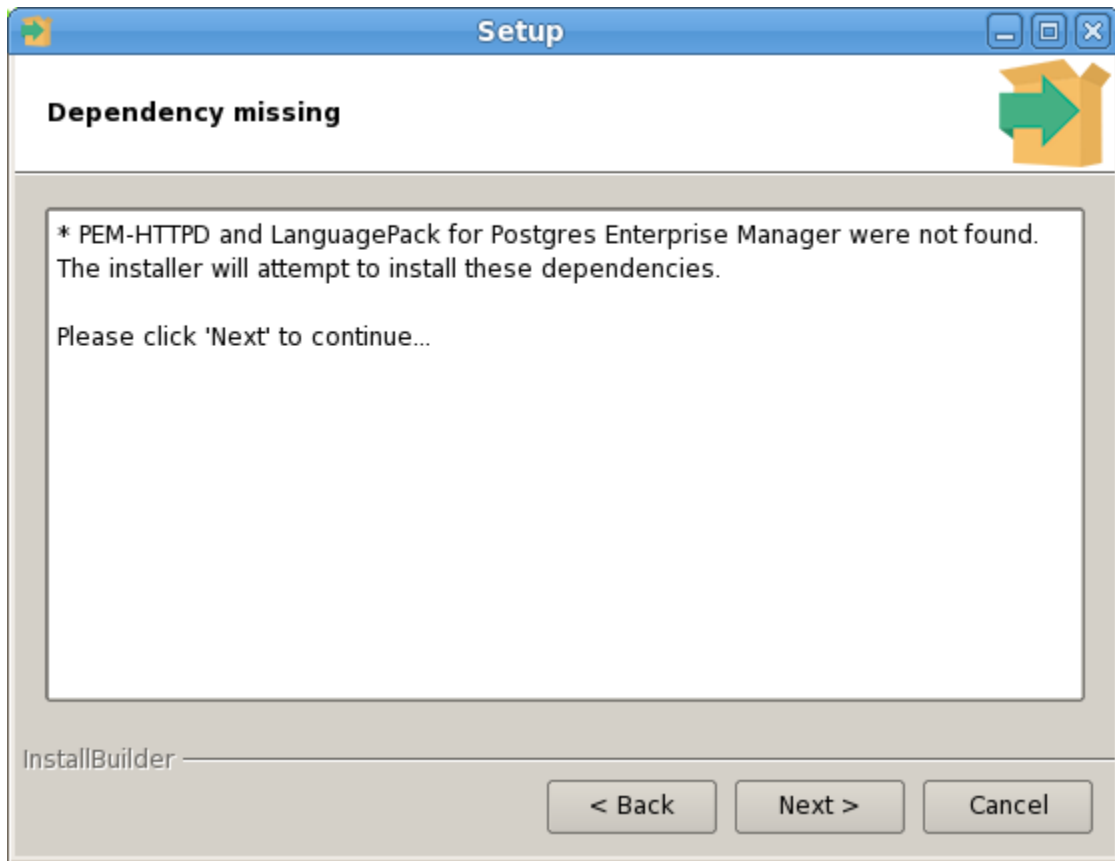


Figure 2.10 - The installation requires dependency upgrades.

Before upgrading the PEM server, the wizard will confirm that the requirements of the new PEM server are present. If any supporting components are missing, or are a version that will not support the new PEM installation, the PEM installation wizard will inform you that it must upgrade the dependencies, and will invoke the required installers (see Figure 2.10).

When the installation wizards complete the dependency upgrades, focus will return to the PEM server setup wizard.

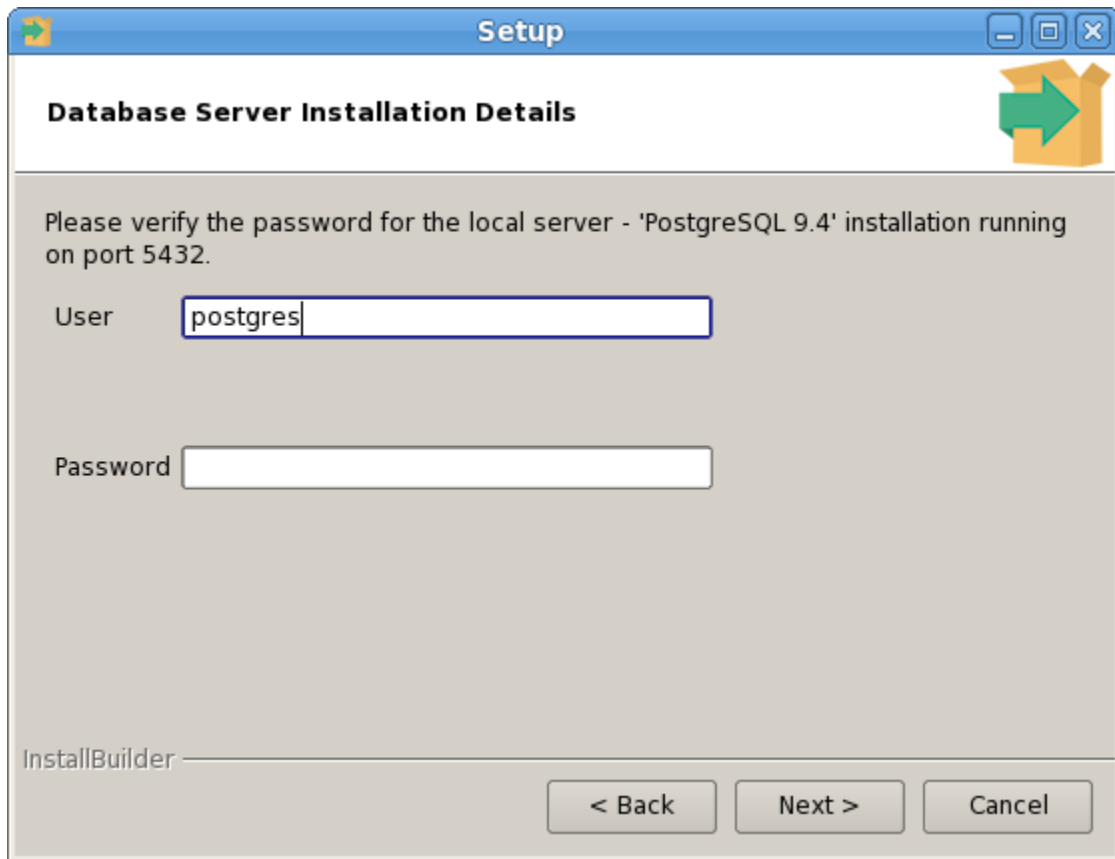


Figure 2.11 - Provide connection information for the backing database.

The wizard then opens the `Database Server Installation Details` dialog, prompting you for connection credentials for database superuser of the PEM backing database (see Figure 2.11). Provide:

- The name of the database superuser in the `User` field.
- The password associated with the database superuser in the `Password` field.

Click `Next` to continue.

The `pemAgent` service account dialog may prompt you for the password of the account under which the PEM agent service runs. If prompted, provide the password, and click `Next` to continue.

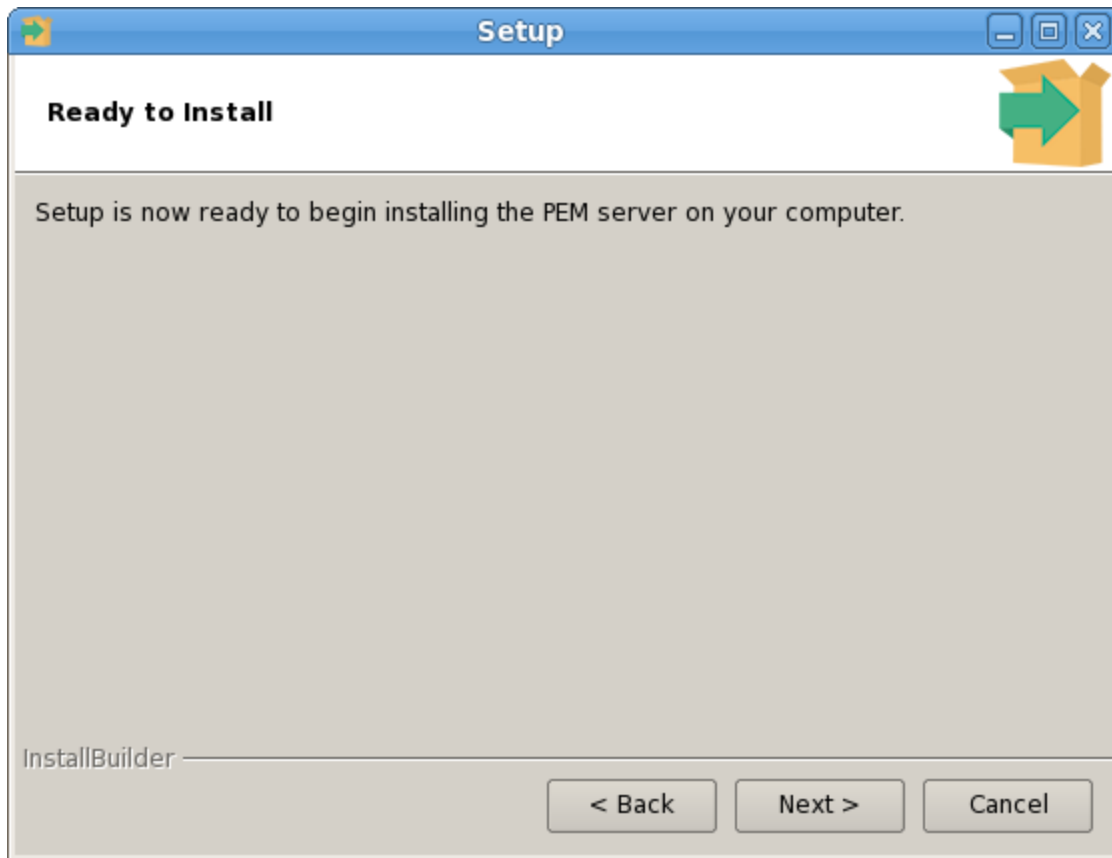


Figure 2.12 - The PEM Setup Wizard is ready to install the PEM server.

The `Ready to Install` dialog will inform you that the setup wizard is ready to perform the installation. Click `Next` to start the installation (see Figure 2.12).

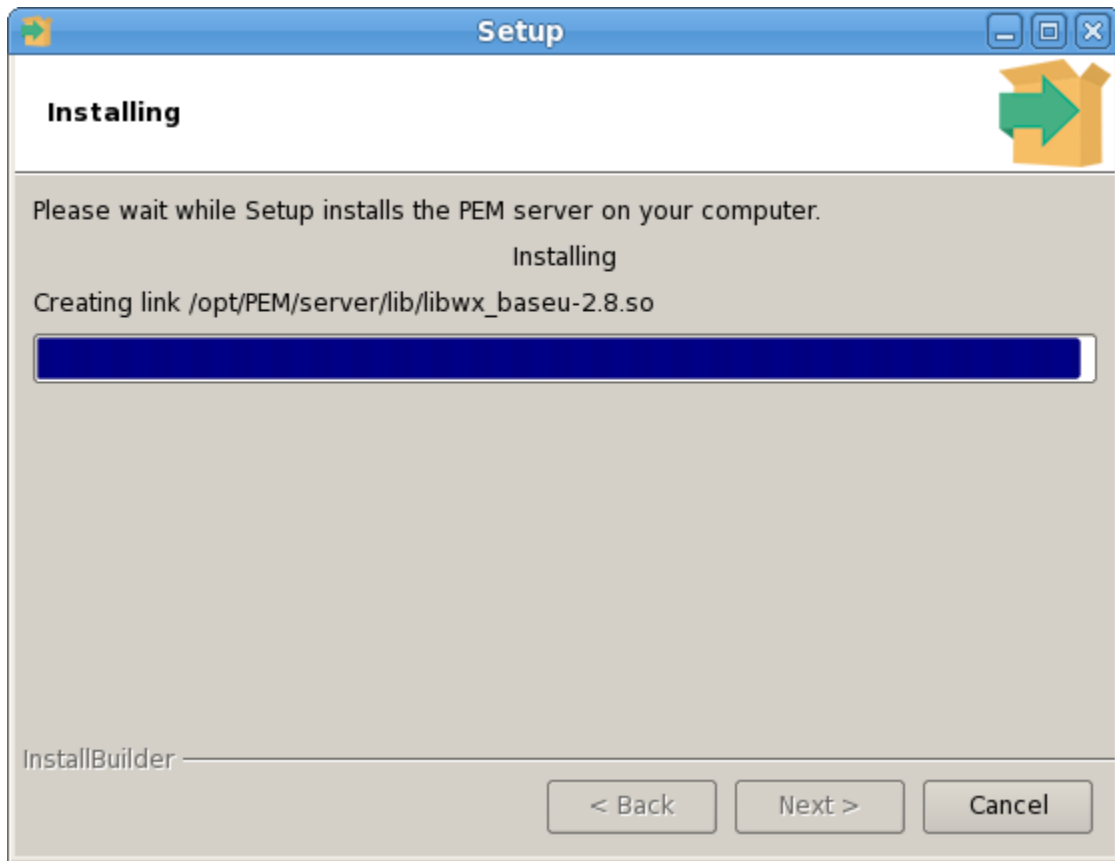


Figure 2.13 - The PEM Server installation in progress.

During the installation, progress bars will keep you informed of the progress of the update (see Figure 2.13).

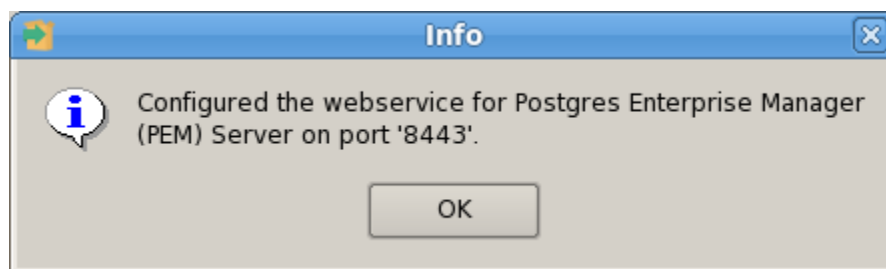


Figure 2.14 - The setup wizard configures the PEM webservice.

After updating the PEM server (and the agent that resides on the same host as the PEM server) and configuring the webservice, the PEM setup wizard notifies you of the port on which the service is listening (see Figure 2.14). Use this port number when connecting to the PEM Server with the PEM client.

Click **OK** to close the **Info** popup. The PEM server setup wizard informs you that the installation is complete (see Figure 2.15).



Figure 2.15 - The PEM Server upgrade is complete.

After upgrading the PEM server, you may wish to upgrade the backing database to a more recent version; for information about upgrading the backing database, see [Section 3](#).

2.1.1 Using an RPM Package to Upgrade the PEM Server

If you installed your PEM server with an RPM package, you can use an RPM to upgrade your PEM server. Please note that if you used a graphical installer to install your existing PEM server, you must remove the existing server before using an RPM package to install PEM 7.5 or later versions. For detailed information about using an RPM package to install PEM, please consult the *PEM Installation Guide*, available at:

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

To use an RPM package to upgrade an existing RPM installation, you must:

1. Use your package manager to upgrade the installed version of the PEM server:

```
yum upgrade edb-pem edb-pem-server
```

2. If used, set SELinux to permissive mode:

```
setenforce 0
```

3. Configure the upgraded server:

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

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

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

If you do not provide configuration properties on the command line, you will be prompted for values by the script. To view script-related help, use the command:

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

After executing the PEM server configuration file, use your version-specific service control command to restart the `httpd` service.

For detailed information about using an RPM package to install or configure the PEM server, please see the *PEM Installation Guide* or the *Advanced Server Installation Guide*, available at:

<http://www.enterprisedb.com/products-services-training/products/documentation/enterpriseedition>

2.3 Upgrading SQL Profiler

The steps required to upgrade SQL Profiler are platform-specific; please see the section for your platform for detailed information.

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 that was installed with a graphical installer, the script resides in the `share/contrib` directory under the Advanced Server installation.

If you are using a PostgreSQL installation on a Linux host, the script resides in the `share/postgresql/contrib` directory under the PostgreSQL installation.

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.

On Windows, you can use the `Services` dialog to control the service. To open the `Services` dialog, navigate through the Control Panel to the System and Security menu. Select `Administrative Tools`, and then double-click the `Services` icon. When the `Services` dialog opens, highlight the service name in the list, and use the option provided on the dialog to `Stop` the service.

After stopping the Postgres Server:

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

By default, the script resides in the `share\contrib` directory under your Advanced Server or PostgreSQL installation.

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. After updating the PEM components, you are ready to update the backing database.

3 Upgrading the Backing Postgres Database

If you are updating both PEM components and the PEM backing database, you should perform PEM component updates (the server, agents and client) before updating the backing database. For more information about updating PEM component software, see section [2](#).

The update process described in this section uses the `pg_upgrade` utility to migrate from one version of the backing server to a more recent version. `pg_upgrade` facilitates migration between any version of Postgres (version 9.3 or later), and any subsequent release of Postgres that is supported on the same platform.

`pg_upgrade` supports a transfer of data between servers of the same type. For example, you can use `pg_upgrade` to move data from a PostgreSQL 9.6 backing database to a PostgreSQL 10 backing database, but not to an Advanced Server 10 backing database. If you wish to migrate to a different type of backing database (i.e from a PostgreSQL server to Advanced Server), please see Section [4](#) for details.

You can find more information about using `pg_upgrade` at:

<http://www.postgresql.org/docs/current/static/pgupgrade.html>

Step 1 - Download and Invoke the Updated Backing Database Installer

Installers for PostgreSQL and Advanced Server are available through the EnterpriseDB website:

<http://www.enterprisedb.com>

After downloading the installer for the server version to which you will be upgrading, invoke the installer on the host of the PEM server. Follow the onscreen instructions of the installation wizard to configure and install the Postgres server.

You can optionally use a custom-built PostgreSQL server as a host of the PEM backing database. Note that if you are upgrading from a PostgreSQL backing database listening on port 5432, the new server must be configured to listen on a different port.

Step 2 - Configure the SSL Utilities on the New Server

The new backing database must be running the same version of `sslutils` that the current backing database is running; you can download the SSL Utils package that is used in EnterpriseDB installers at:

<http://www.enterprisedb.com/downloads/component-source-code>

You are *not* required to manually add the `sslutils` extension when using the Advanced Server as the new backing database. The process of configuring `sslutils` is platform-specific.

On Linux

If you are using Linux, you can download versions of the archived SSL Utils file from:

<http://www.enterprisedb.com/downloads/component-source-code>

When the download completes, extract the `sslutils` folder, and move it into the Postgres installation directory for the Postgres version to which you are upgrading.

Open a command line, assume superuser privileges, and set the value of the `PATH` environment variable to allow `make` to locate the `pg_config` program:

```
export PATH=$PATH:/opt/Postgres/x.x/bin/
```

Where:

Postgres specifies either:

- PostgreSQL if you are upgrading to a PostgreSQL server.
- PostgresPlus if you are upgrading to an Advanced Server server.

x.x specifies the version of Postgres to which you are migrating.

Then, use `yum` to install `sslutil` dependencies:

```
yum install openssl-devel
```

Navigate into the `sslutils` folder, and build the `sslutils` package by entering:

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

On Windows

`sslutils` must be compiled on the new backing database with the same compiler that was used to compile `sslutils` on the original backing database. If you are moving to a Postgres database that was installed using a PostgreSQL one-click installer (from EnterpriseDB) or an Advanced Server installer, use Visual Studio to build `sslutils`. If you are upgrading to:

- PostgreSQL 9.3 or later, use Visual Studio 2010

For detailed information about building a specific version of Postgres on Windows, please consult the core documentation for that version. Core documentation is available at the PostgreSQL project website at:

<http://www.postgresql.org/docs/>

or at the EnterpriseDB website at:

<http://www.enterprisedb.com/products-services-training/products/documentation/enterpriseedition>

While specific details of the process will vary by platform and compiler, the basic steps on each platform are the same. The example that follows demonstrates compiling OpenSSL support for PostgreSQL on a 32-bit Windows system.

Before compiling the OpenSSL extension, you must locate and install OpenSSL for your version of Windows. Before invoking the OpenSSL installer you may be required to download and install a pre-requisite redistributable (such as `vc redistrib_x86.exe`).

After installing OpenSSL, download and unpack the `SSL Util` utility package available at:

<http://www.enterprisedb.com/downloads/component-source-code>

Copy the unpacked `sslutils` utilities folder to the Postgres installation directory (i.e. `C:\Program Files\PostgreSQL\9.x`).

Open the Visual Studio command line, and navigate into the `sslutils` directory. Use the following commands to build `sslutils`:

```
SET USE_PGXS=1
SET GETTEXT_PATH=path_to_gettext
SET OPENSSL_PATH=path_to_openssl
SET PG_PATH=path_to_pg_installation_dir
SET ARCH=x86
msbuild sslutils.proj /p:Configuration=Release
```

Where:

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

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

path_to_pg_installation_dir specifies the location of the Postgres installation.

For example, the following set of commands builds OpenSSL support into the PostgreSQL 10 server:

```
SET USE_PGXS=1
SET OPENSSELPATH=C:\OpenSSL-Win32
SET GETTEXT_PATH="C:\Program Files\PostgreSQL\10"
SET PGPATH="C:\Program Files\PostgreSQL\10"
SET ARCH=x86
msbuild sslutils.proj /p:Configuration=Release
```

When the build completes, the `sslutils` directory will contain the following files:

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

Copy the compiled `sslutils` files to the appropriate directory for your installation; for example:

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

Step 3 - Stop the Services

Stop the services of both the old backing database and the new backing database.

On RHEL or CentOS 6.x, open a command line and assume the identity of a superuser. Enter the command:

```
/etc/init.d/service_name stop
```

On RHEL or CentOS 7.x, open a command line and assume the identity of a superuser. Enter the command:

```
systemctl/service_name stop
```

Where *service_name* specifies the name of the Postgres service.

On Windows, you can use the `Services` dialog to control the service. To open the `Services` dialog, navigate through the Control Panel to the System and Security menu. Select `Administrative Tools`, and then double-click the `Services` icon. When the `Services` dialog opens, highlight the service name in the list, and use the option provided on the dialog to Stop the service.

Step 4 - Use `pg_upgrade` to update the Server

You can use the `pg_upgrade` utility to perform an in-place transfer of existing data between the old backing database and the new backing database. If your server is configured to enforce `md5` authentication, you may need to add an entry to the `.pgpass` file that specifies the connection properties (and password) for the database superuser, or modify the `pg_hba.conf` file to allow `trust` connections before invoking `pg_upgrade`. For more information about creating an entry in the `.pgpass` file, please see the PostgreSQL core documentation, available at:

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

During the upgrade process, `pg_upgrade` will write a series of log files. The cluster owner must invoke `pg_upgrade` from a directory in which they have write privileges. If the upgrade completes successfully, `pg_upgrade` will remove the log files when the upgrade completes. To instruct `pg_upgrade` to not delete the upgrade log files, include the `--retain` keyword when invoking `pg_upgrade`.

To invoke `pg_upgrade`, assume the identity of the cluster owner, navigate into a directory in which the cluster owner has write privileges, and execute the command:

```
path_to_pg_upgrade/pg_upgrade
-d old_data_dir_path
-D new_data_dir_path
-b old_bin_dir_path -B new_bin_dir_path
-p old_port -P new_port
-u user_name
```

Where:

path_to_pg_upgrade specifies the location of the `pg_upgrade` utility. By default, `pg_upgrade` is installed in the `bin` directory under your Postgres directory.

old_data_dir_path specifies the complete path to the `data` directory of the old backing database.

new_data_dir_path specifies the complete path to the `data` directory of the new backing database.

old_bin_dir_path specifies the complete path to the `bin` directory of the old backing database.

new_bin_dir_path specifies the complete path to the `bin` directory of the old backing database.

old_port specifies the `port` on which the old server is listening.

new_port specifies the `port` on which the new server is listening.

user_name specifies the name of the cluster owner.

For example, the following command:

```
C:\>"C:\Program Files\PostgreSQL\10\bin\pg_upgrade.exe"
-d "C:\Program Files\PostgreSQL\9.6\data"
-D "C:\Program Files\PostgreSQL\10\data"
-b "C:\Program Files\PostgreSQL\9.6\bin"
-B "C:\Program Files\PostgreSQL\10\bin"
-p 5432 -P 5433
-u postgres
```

Instructs `pg_upgrade` to migrate the PEM database from PostgreSQL 9.6 to PostgreSQL 10 on a Windows system (if the backing databases are installed in their default locations).

Once invoked, `pg_upgrade` will perform consistency checks before moving the data to the new backing database. When the upgrade is finished, `pg_upgrade` will notify you that the upgrade is complete.

For detailed information about using `pg_upgrade` options, or troubleshooting the upgrade process, please see:

<http://www.postgresql.org/docs/current/static/pgupgrade.html>

Step 5 - Copy the Certificate Files from the Old Database to the New Database

Copy the following certificate files from the `data` directory of the old backing database to the `data` directory of the new backing database:

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

By default, the files are in the `data` directory, under your Postgres installation.

Once in place on the target server, the files should have the (platform-specific) permissions described below:

Permissions and Ownership on Linux

File Name	Owner	Permissions
<code>ca_certificate.crt</code>	<code>postgres</code>	<code>-rw-----</code>
<code>ca_key.key</code>	<code>postgres</code>	<code>-rw-----</code>
<code>root.crt</code>	<code>postgres</code>	<code>-rw-----</code>
<code>root.crl</code>	<code>postgres</code>	<code>-rw-----</code>
<code>server.key</code>	<code>postgres</code>	<code>-rw-----</code>
<code>server.crt</code>	<code>postgres</code>	<code>-rw-r--r--</code>

On Linux, the certificate files must be owned by `postgres`. You can use the following command at the command line to modify the ownership of the files:

```
chown postgres file_name
```

Where `file_name` specifies the name of the certificate file.

The `server.crt` file may only be modified by the owner of the file, but may be read by any user. You can use the following command to set the file permissions for the `server.crt` file:

```
chmod 644 server.crt
```

The other certificate files may only be modified or read by the owner of the file. You can use the following command to set the file permissions:

```
chmod 600 file_name
```

Where `file_name` specifies the name of the file.

Permissions and Ownership on Windows

On Windows, the certificate files moved from the source host must be owned by the service account that performed the PEM server and backing database installation on the target host. If you invoked the PEM server and Postgres installer using the `Run as Administrator` option (selected from the context menu of the installer), the owner of the certificate files will be `Administrators`.

To review and modify file permissions on Windows, right-click on the file name, and select `Properties`.

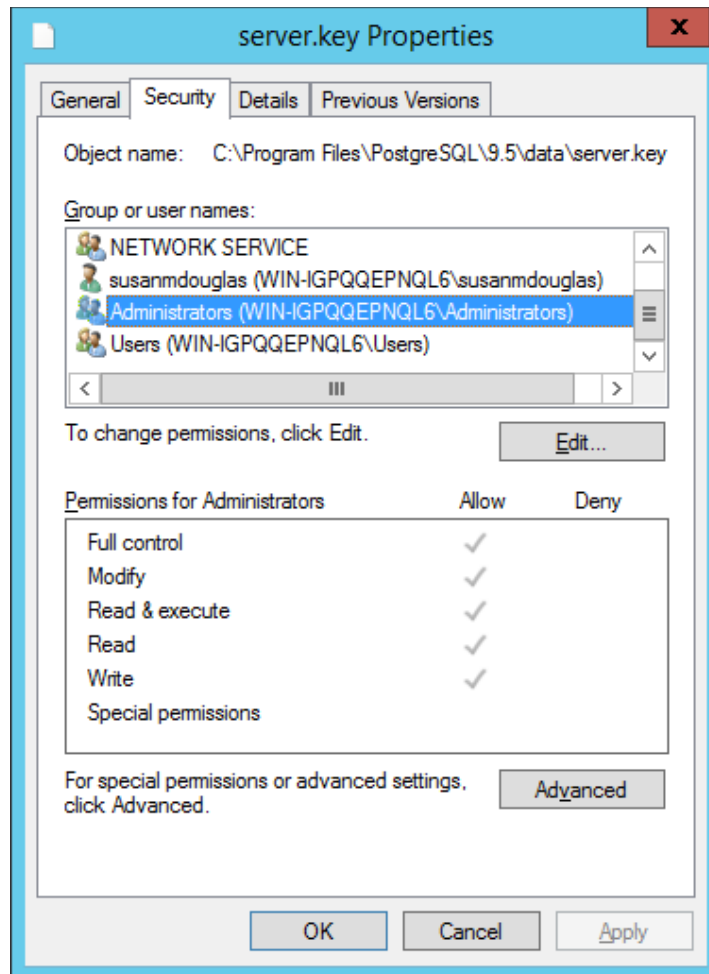


Figure 3.3 - The Security tab.

Navigate to the `Security` tab (see Figure 3.3) and highlight a `Group` or `user name` to view the assigned permissions. Select `Edit` or `Advanced` to access dialogs that allow you to modify the permissions associated with the selected user.

Step 6 - Update the New Server Configuration File

The `postgresql.conf` file contains parameter settings that specify server behavior. You will need to modify the `postgresql.conf` file on the new server to match the configuration specified in the `postgresql.conf` file of the old server.

By default, the `postgresql.conf` file is located:

- For Postgres version lower than 10 on Linux, in `/opt/PostgreSQL/version.x/data`
- For Postgres version 10 or higher when installed with graphical installers on Linux, in `/opt/PostgreSQL/version/data`
- For Postgres version 10 or higher when installed with an RPM on Linux, in `/usr/edb/PostgreSQL/version/data`
- For any Postgres version on Windows, in `C:\Program Files\PostgreSQL\version.x\data`

Where, `version` is the major version of Postgres on your system.

Use your choice of editor to update the `postgresql.conf` file of the new server. Modify the following parameters:

The `port` parameter to listen on the port monitored by your original backing database (typically, 5432).

The `ssl` parameter should be set to `on`.

You must also ensure that the following parameters are enabled. If the parameters are commented out, remove the pound sign from in front of each `postgresql.conf` file entry:

```
ssl_cert_file = 'server.crt'           # (change requires restart)
ssl_key_file = 'server.key'           # (change requires restart)
ssl_ca_file = 'root.crt'              # (change requires restart)
ssl_crl_file = 'root.crl'
```

Your installation may have other parameter settings that require modification to ensure that the new backing database behaves in a manner comparable to the old backing database. Review the `postgresql.conf` files carefully to ensure that the configuration of the new server matches the configuration of the old server.

Step 7 - Update the New Server Authentication File

The `pg_hba.conf` file contains parameter settings that specify how the server will enforce host-based authentication. When you install the PEM server, the installer modifies the `pg_hba.conf` file, adding entries to the top of the file:

```
# Adding entries for PEM agents and admins to connect to
PEM server
hostssl pem +pem_user 192.168.2.0/24 md5
hostssl pem +pem_agent 192.168.2.0/24 cert
# Adding entries (localhost) for PEM 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
```

By default, the `pg_hba.conf` file is located:

On Linux, in `/opt/PostgreSQL/10.x/data`

On Windows, in `C:\Program Files\PostgreSQL\10.x\data`

By default, the `pg_hba.conf` file is located:

- For Postgres version lower than 10 on Linux, in:
`/opt/PostgreSQL/version.x/data`
- For Postgres version 10 or higher when installed with graphical installers on Linux, in:
`/Opt/PostgreSQL/version/data`
- For Postgres version 10 or higher when installed with RPMs on Linux, in
`/var/lib/PostgreSQL/version/data`
- For Advanced Server version 10 or higher when installed with RPMs on Linux, in:
`/var/lib/edb/ASversion/data`
- For any Postgres version on Windows, in:
`C:\Program Files\PostgreSQL\version.x\data`

Where, `version` is the major version of Postgres on your system.

Using your editor of choice, copy the entries from the `pg_hba.conf` file of the old server to the `pg_hba.conf` file for the new server.

Step 8 - Restart the New Postgres Server

Start the service of the new backing database. On RHEL or CentOS 6.x, open a command line and assume the identity of a superuser. Enter the command:

```
/etc/init.d/service_name start
```

On RHEL or CentOS 7.x, open a command line and assume the identity of a superuser. Enter the command:

```
systemctl stop service_name
```

Where *service_name* is the name of the backing database server.

If you are using Windows, you can use the `Services` dialog to control the service. To open the `Services` dialog, navigate through the Control Panel to the System and Security menu. Select Administrative Tools, and then double-click the `Services` icon. When the `Services` dialog opens, highlight the service name in the list, and use the option provided on the dialog to Start the service.

4 Moving the Postgres Enterprise Manager™ Server

The steps in this section describe how to move a PEM server from one host machine to a new host machine. The PEM server on the new host (the target) must be installed with the same version of the PEM server installer as the original host (the source). Please note that if you do not use the same installer version, you may encounter a schema-mismatch error.

The backing database of the target server (either PostgreSQL or Advanced Server) may be of the same type and version, or a different type and version than the backing database of the source PEM server. A PEM server that resides on a PostgreSQL host can be migrated to an Advanced Server host, or vice versa.

Before starting the server migration, you should ensure that the firewalls between the source host, the target host, and the host of any PEM agent will allow connections between the services.

Step One - Prepare the Target Host

Invoke the installer for the PEM server on the target host. Please note that you must use the same version of the PEM server installer that you used when installing the source PEM server.

The backing database of the target server may be a different version or type than the backing database of the source. If the new PEM server does *not* reside on the same type of backing database as the original server, you must ensure that the same version of the `sslutils` extension is installed on the new server host. The version of `sslutils` that is distributed with the PEM installers is freely available for download from the EnterpriseDB website at:

<http://www.enterprisedb.com/downloads/component-source-code>

For information about installing the PEM server or the `sslutils` extension, please refer to the PEM Installation Guide, available at:

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

Step Two – Drop Existing Schemas from the New PEM Server

The migration process re-creates the `pem`, `pemdata`, and `pemhistory` schemas from the source PEM server on the target PEM server. In preparation for the move, use the `psql`

client to delete these schemas from the `pem` database on the target host. You can open the `psql` client at the command line, or by selecting `SQL Shell (psql)` from the `Postgres Enterprise Manager` menu.

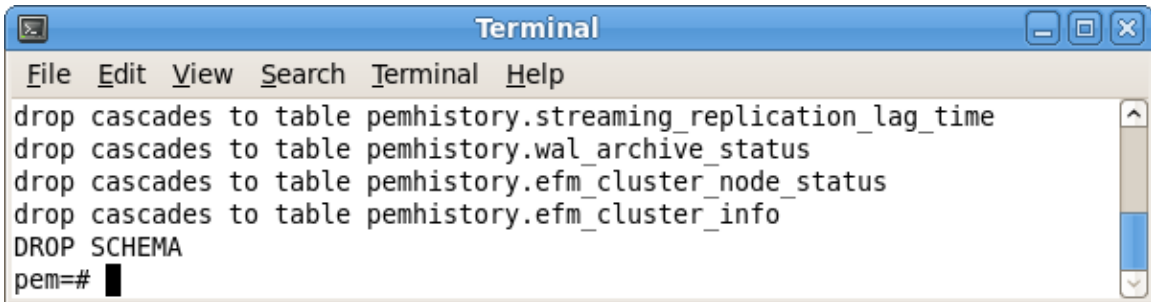
When the `psql` client opens, connect to the `pem` backing database as the database superuser. After connecting to the `pem` database on the target host, use the following commands to drop the schemas:

```
DROP SCHEMA pem CASCADE;

DROP SCHEMA pemdata CASCADE;

DROP SCHEMA pemhistory CASCADE;
```

When dropping the schemas, you must include the `CASCADE` keyword, instructing the server to delete all dependent objects. When executing the command, the `psql` client displays a list of the dependent objects; the client confirms each the schema is removed by displaying `DROP SCHEMA` (as shown in Figure 4.1).



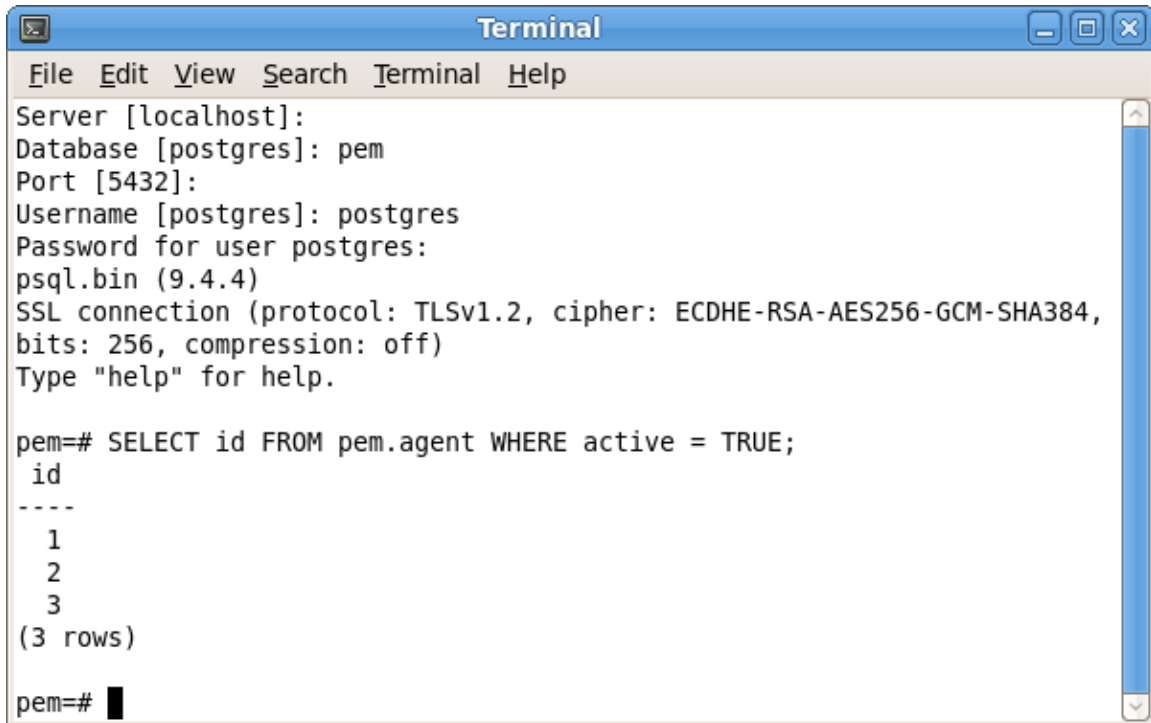
```
Terminal
File Edit View Search Terminal Help
drop cascades to table pemhistory.streaming_replication_lag_time
drop cascades to table pemhistory.wal_archive_status
drop cascades to table pemhistory.efm_cluster_node_status
drop cascades to table pemhistory.efm_cluster_info
DROP SCHEMA
pem=#
```

Figure 4.1 - Dropping the pem schema.

Step Three - Prepare the PEM Agents on the New PEM Server

Before moving the PEM server, you must identify the number of agents that are monitored by the source PEM server, and create identities for that number of agents (less one) on the target server. To discover the total number of PEM agents monitored by the PEM server, connect to the `pem` database on the source host with the `psql` client, and query the `pem.agent` table (as shown in Figure 4.2).

```
SELECT id FROM pem.agent WHERE active = true;
```



```

Terminal
File Edit View Search Terminal Help
Server [localhost]:
Database [postgres]: pem
Port [5432]:
Username [postgres]: postgres
Password for user postgres:
psql.bin (9.4.4)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384,
bits: 256, compression: off)
Type "help" for help.

pem=# SELECT id FROM pem.agent WHERE active = TRUE;
 id
----
  1
  2
  3
(3 rows)

pem=# █

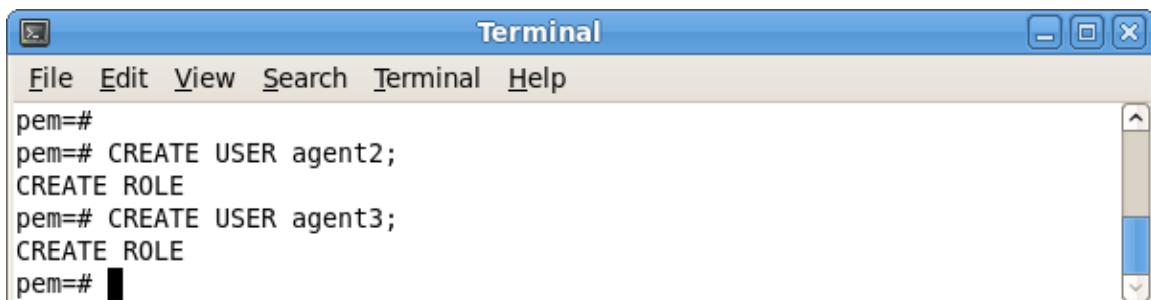
```

Figure 4.2 - Querying the pem database on the source PEM server.

You must manually create the number of agents that reside on the original PEM server, less one; the PEM server installer has already created one agent on the target host. For example, if the source server contains three agents, you must manually create two additional agents. Open a `psql` session with the `pem` database on the target server, and create the required agents. Use the command:

```
CREATE USER agentx;
```

Where `x` specifies an agent number (see Figure 4.3). Remember, `agent1` is created on the target host by the PEM server installer.



```

Terminal
File Edit View Search Terminal Help
pem=#
pem=# CREATE USER agent2;
CREATE ROLE
pem=# CREATE USER agent3;
CREATE ROLE
pem=# █

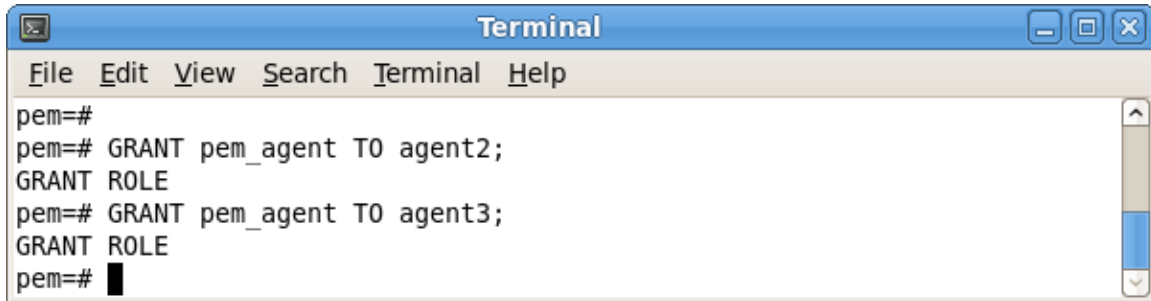
```

Figure 4.3 - Creating additional agents on the target.

Then, use the `GRANT` command to assign each agent that resides on the target PEM server `pem_agent` permissions:

```
GRANT pem_agent TO agentx;
```

Where `x` specifies an agent number (see Figure 4.4).



```
Terminal
File Edit View Search Terminal Help
pem=#
pem=# GRANT pem_agent TO agent2;
GRANT ROLE
pem=# GRANT pem_agent TO agent3;
GRANT ROLE
pem=# █
```

Figure 4.4 - Granting privileges to the agents on the target.

Step Four - Generate a Backup Script of the Source PEM Server

You can use the `pg_dump` utility to generate a script that contains the commands required to recreate the `pem` database on the target host. By default, `pg_dump` is installed in the `bin` directory under your Postgres installation. To invoke `pg_dump`, open a command line, navigate to the `bin` directory, and enter:

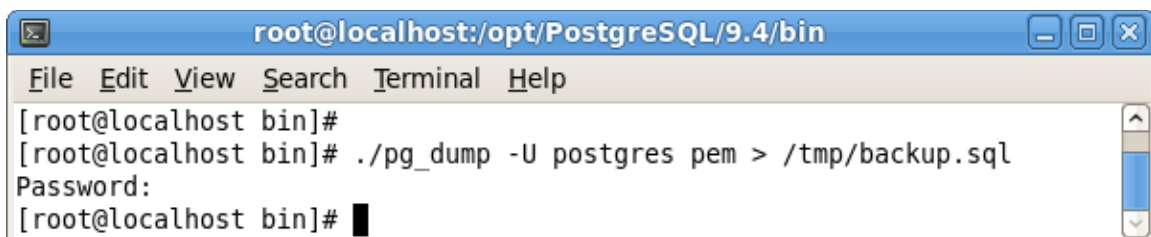
```
pg_dump -U user_name db_name > file_name
```

Where:

`user_name` specifies the name of the database superuser for the PEM backing database.

`file_name` specifies the name of the script generated by `pg_dump`.

When prompted, provide the password associated with the user specified.



```
root@localhost:/opt/PostgreSQL/9.4/bin
File Edit View Search Terminal Help
[root@localhost bin]#
[root@localhost bin]# ./pg_dump -U postgres pem > /tmp/backup.sql
Password:
[root@localhost bin]# █
```

Figure 4.5 - Generating a backup script.

The command shown in Figure 4.5 instructs `pg_dump` to generate a script that (when executed) will re-create the `pem` database. The script will be named `backup.sql`, and will be created in the `tmp` directory. `pg_dump` is connecting to the server using the credentials of the user, `postgres`.

Note that invoking the `pg_dump` utility will not interrupt current database users.

Step Five - Move the Backup to the Target Host

Move the script generated by the `pg_dump` utility to the target host of the PEM server.

Step Six - Restore the Backup on the Target Host

Open a command line on the target host and navigate into the `bin` directory (under the Postgres backing database installation directory). Start `psql`, executing the script generated by the `pg_dump` utility:

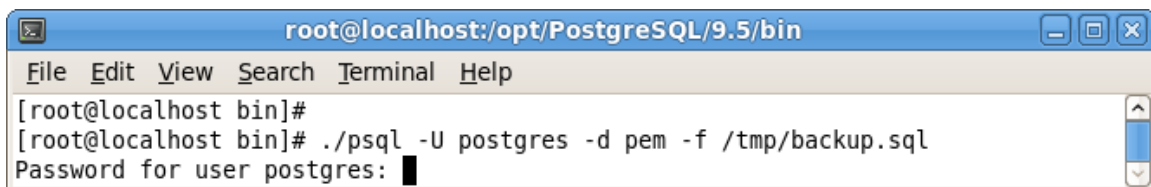
```
psql -U user_name -d pem -f file_name
```

Where:

`user_name` specifies the name of the database superuser. The user specified must have connection privileges for the backing database.

`file_name` specifies the complete path to the backup script generated by `pg_dump`.

When prompted, provide the password associated with the database superuser.



```

root@localhost:/opt/PostgreSQL/9.5/bin
File Edit View Search Terminal Help
[root@localhost bin]#
[root@localhost bin]# ./psql -U postgres -d pem -f /tmp/backup.sql
Password for user postgres: █

```

Figure 4.6 - Restoring a backup script.

The example shown in Figure 4.6 uses the `psql` client to invoke a script named `backup.sql` to recreate the `pem` database. The script is invoked using the privileges associated with the database superuser, `postgres`.

Step Seven - Stop the Database Server on the Target Host

To stop the PEM server on Linux, use the command:

```
/etc/init.d/service_name stop
```

service_name specifies the name of the backing database server. For a PostgreSQL backing database, the service name is `postgresql-x.x`, and for an Advanced Server backing database, the service name is `ppas-x.x`, where *x* specifies the version number.

If you are using Windows, you can use the `Services` dialog to control the service. To open the `Services` dialog, navigate through the Control Panel to the System and Security menu. Select Administrative Tools, and then double-click the `Services` icon. When the `Services` dialog opens, highlight the service name in the list, and use the option provided on the dialog to Stop the service.

Step Eight - Copy the Certificate Files to the Target Host

You must replace the certificate files that are created when the target host is installed with the certificate files of the source host. Copy the following files from the source PEM server to the target PEM server:

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

Copy the files to the `data` directory under the Postgres installation that provides the backing database for the target cluster. On Linux, by default, the files reside in:

```
/opt/PostgreSQL/x.x/data/
```

On Windows, the files reside in:

```
C:\Program Files\PostgreSQL\x.x\data
```

The files will already exist on the target cluster; delete the existing files before performing the copy, or overwrite the existing files with the files from the source server. Once in place on the target server, the files should have the (platform-specific) permissions described in the sections that follow.

Permissions and Ownership on Linux

File Name	Owner	Permissions
ca_certificate.crt	postgres	-rw-----
ca_key.key	postgres	-rw-----
root.crt	postgres	-rw-----
root.crl	postgres	-rw-----
server.key	postgres	-rw-----
server.crt	postgres	-rw-r--r--

On Linux, the certificate files must be owned by `postgres`. You can use the following command at the command line to modify the ownership of the files:

```
chown postgres file_name
```

Where *file_name* specifies the name of the certificate file.

The `server.crt` file may only be modified by the owner of the file, but may be read by any user. You can use the following command to set the file permissions for the `server.crt` file:

```
chmod 644 server.crt
```

The other certificate files may only be modified or read by the owner of the file. You can use the following command to set the file permissions:

```
chmod 600 file_name
```

Where *file_name* specifies the name of the file.

Permissions and Ownership on Windows

On Windows, the certificate files moved from the source host must be owned by the service account that performed the PEM server and backing database installation on the target host. If you invoked the PEM server and Postgres installer using the `Run as Administrator` option (selected from the context menu of the installer), the owner of the certificate files will be `Administrators`.

To review and modify file permissions on Windows, right-click on the file name, and select `Properties`.

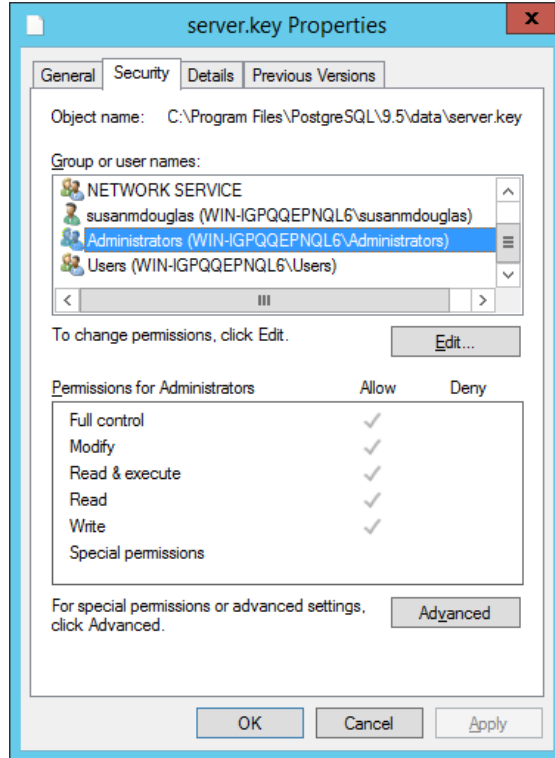


Figure 4.7 - The Permissions tab.

Navigate to the Security tab (see Figure 4.7) and highlight a Group or user name to view the assigned permissions. Select Edit or Advanced to access dialogs that allow you to modify the permissions associated with the selected user.

Step Nine - Move the PEM Agent Certificate Files to the PEM Server Host

You must move the certificate files used by the PEM agent of the source PEM server to the target host. This step is platform-specific.

On Linux

Copy the `agent1.key` and `agent1.crt` files from the source host to the target host. By default, on Linux, the files are installed in `/root/.pem`; copy the files to the same directory on the target host.

File ownership and permissions of the files must be set to:

File Name	Owner	Permissions
<code>agent1.key</code>	root	-rw-----
<code>agent1.crt</code>	root	-rw-r--r--

If necessary, navigate to `/root/.pem`, and use the following commands to modify the permissions and ownership of the `agent1.key` file:

```
chmod 600 agent1.key
chown root agent1.key
```

Use the following commands to modify the permissions and ownership of the `agent1.crt` file:

```
chmod 644 agent1.crt
chown root agent1.crt
```

On Windows

Copy the `agent1.key` and `agent1.crt` files from the source host to the target host. On Windows, the files are located in:

```
C:\Users\user_name\AppData\Roaming\pem
```

Where *user_name* is the name of the user that invoked the PEM installer.

The ownership and permissions associated with the certificate files on the target machine should match the ownership and permissions of the certificate files on the source machine. If you invoked the PEM server and Postgres installer using the `Run as Administrator` option (selected from the context menu of the installer), the owner of the agent certificate files will be `Administrators`.

To review and modify file permissions on Windows, right-click on the file name, and select `Properties`. Navigate to the `Security` tab and highlight a `Group` or `user` name to view the assigned permissions. Select `Edit` or `Advanced` to access dialogs that allow you to modify the permissions associated with the selected user.

Step Ten - Update the `pg_hba.conf` Files on the Target Host

Modify the `pg_hba.conf` file on the target host to allow connections from each PEM agent. By default, the `pg_hba.conf` file is located in the `data` directory under your Postgres installation.

Step Eleven - Start the Server on the Target Host

After modifying the `pg_hba.conf` file, you must restart the server for the changes to take effect.

To restart the database server on Linux, use the command:

```
/etc/init.d/service_name start
```

Where *service_name* is the name of the backing database server.

If you are using Windows, you can use the *Services* dialog to control the service. To open the *Services* dialog, navigate through the Control Panel to the System and Security menu. Select *Administrative Tools*, and then double-click the *Services* icon. When the *Services* dialog opens, highlight the service name in the list, and use the option provided on the dialog to *Start* the service.

Step Twelve - Connecting Monitored Agents to the New PEM Server Host

To instruct existing PEM agents to connect to the new PEM server host, you must:

- Ensure that the PEM agent host can connect to the new PEM server host.
- Modify the registry (on each Windows host with a PEM agent) or the agent configuration files (on each Linux host with a PEM agent), specifying the IP address and port of the new PEM server.
- Restart the PEM agent's service.

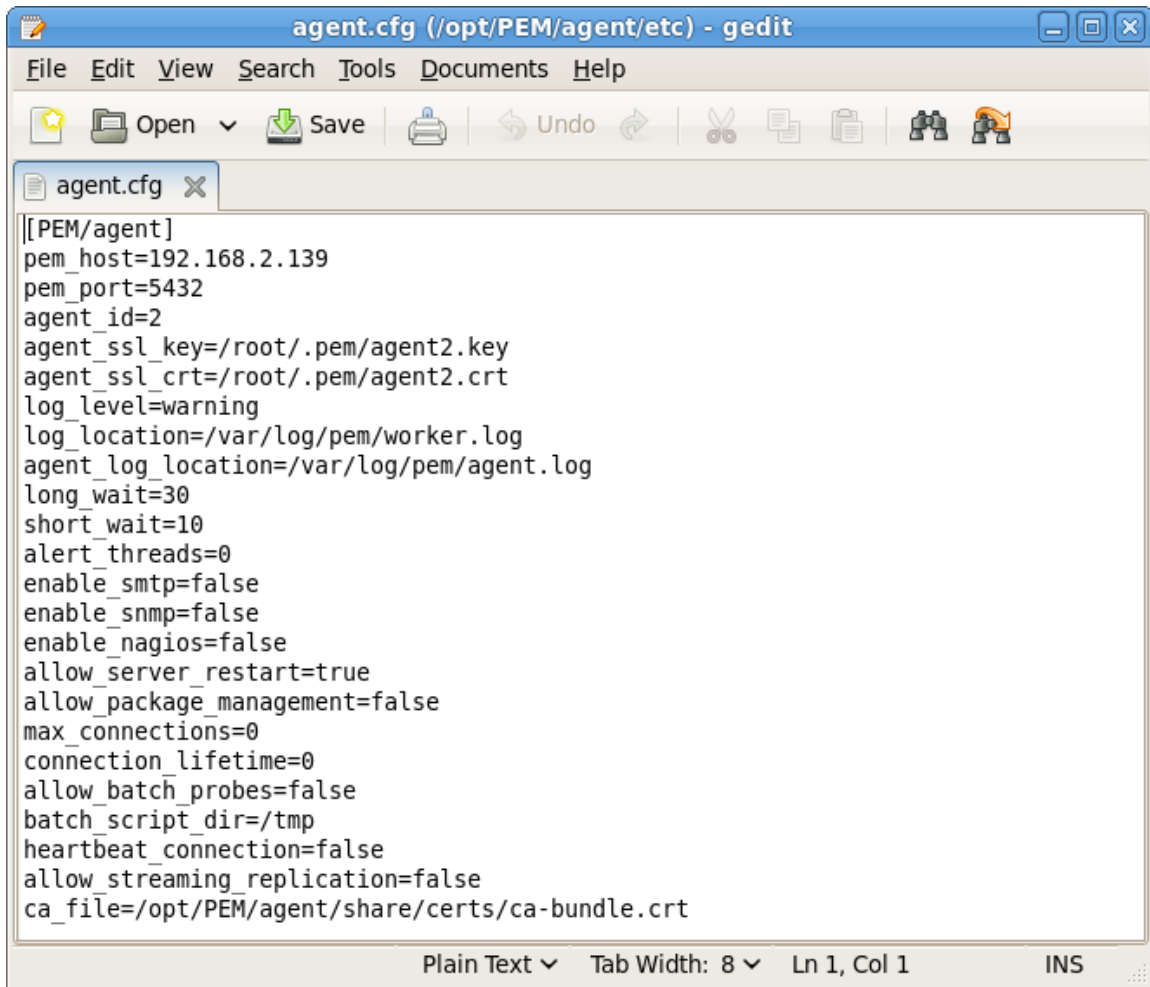
These steps are platform-specific.

If the PEM Agent Resides on Linux

Use your choice of editor to modify the `agent.cfg` file (shown in Figure 4.8), specifying the new IP address and port number of the PEM server in the `pem_host` and `pem_port` parameters.

By default, the `agent.cfg` file is located in:

- `/opt/PEM/agent/etc/agent.cfg`



```

agent.cfg (/opt/PEM/agent/etc) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
agent.cfg x
[[PEM/agent]
pem_host=192.168.2.139
pem_port=5432
agent_id=2
agent_ssl_key=/root/.pem/agent2.key
agent_ssl.crt=/root/.pem/agent2.crt
log_level=warning
log_location=/var/log/pem/worker.log
agent_log_location=/var/log/pem/agent.log
long_wait=30
short_wait=10
alert_threads=0
enable_smtp=false
enable_snmp=false
enable_nagios=false
allow_server_restart=true
allow_package_management=false
max_connections=0
connection_lifetime=0
allow_batch_probes=false
batch_script_dir=/tmp
heartbeat_connection=false
allow_streaming_replication=false
ca_file=/opt/PEM/agent/share/certs/ca-bundle.crt
Plain Text Tab Width: 8 Ln 1, Col 1 INS

```

Figure 4.8 - The agent.cfg file.

After modifying the `agent.cfg` file, you must restart the PEM agent service; you can use the `pemagent` service script on the Linux command line to restart the service:

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

If the PEM Agent Resides on Windows

Before modifying the Windows registry on the monitored node, confirm that the firewall on the host of the PEM agent will allow connections to the PEM server. After confirming that the PEM agent host can connect to the PEM server host, you can use the Windows Registry Editor to review and edit the `PEM_HOST` and `PEM_PORT` entries to ensure that they correctly identify the host and port used by the PEM server. To open the Registry Editor, enter `regedit` in the Windows Run dialog or in the Windows start menu search box.

Navigate through the registry tree control (see Figure 4.9) to view or modify registry entries. On 64-bit Windows, the PEM agent registry entries are located:

HKEY_LOCAL_MACHINE → SOFTWARE → wow6432Mode → EnterpriseDB
 → PEM → agent

On 32-bit Windows, the PEM agent registry entries are located:

HKEY_LOCAL_MACHINE → SOFTWARE → EnterpriseDB → PEM → agent

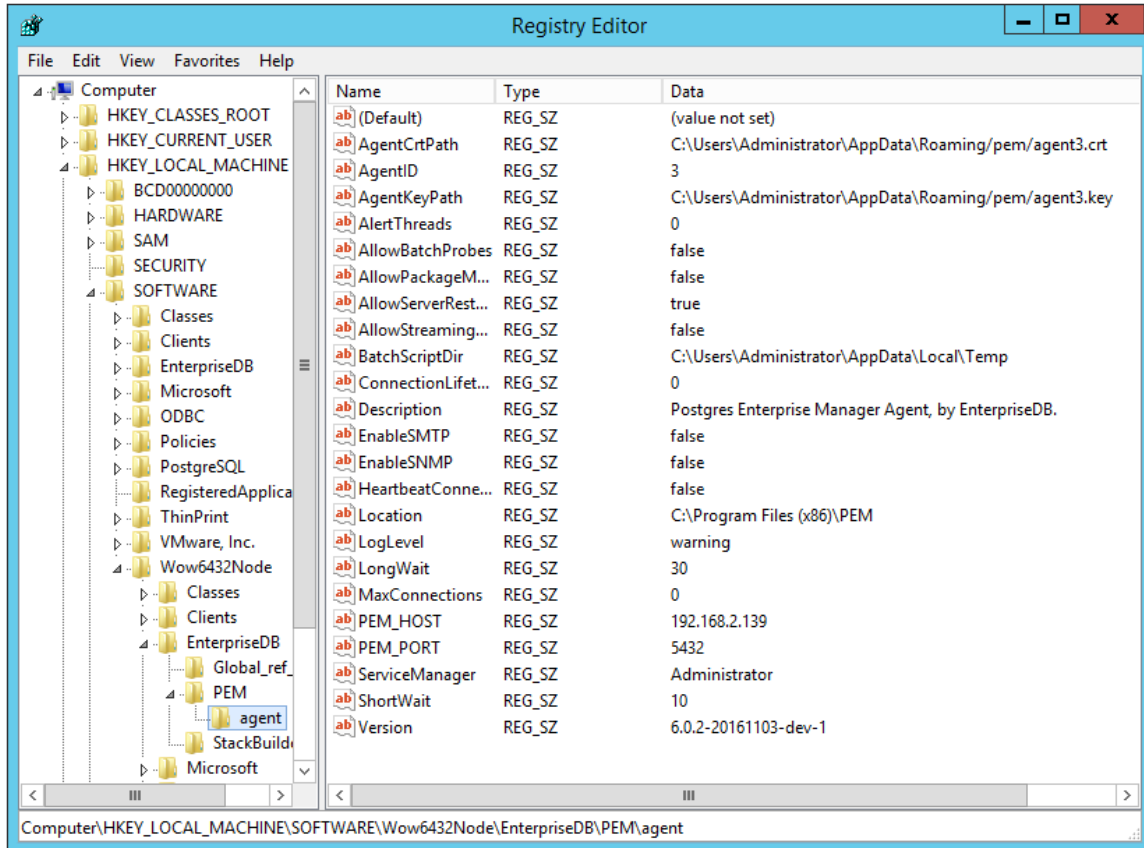


Figure 4.9 - The Windows Registry Editor.

The PEM_HOST and PEM_PORT entries must specify the address and port number of the new PEM server on the target host. To modify a registry entry, right click on the entry Name, and select Modify from the context menu to open the Edit String dialog.

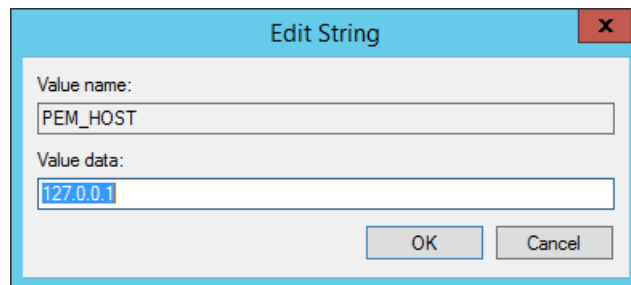


Figure 4.10 - The Windows Registry Editor.

Use the `Edit String` dialog to make any changes to the value of the entry (see Figure 4.10). When you're finished, click `OK` to save your changes, or `Cancel` to exit without saving.

After modifying the registry, you must restart the PEM agent's service; you can use the `Services` dialog (accessed through the `Windows Control Panel`) to restart the `Postgres Enterprise Manager - pemAgent` service (see Figure 4.11).

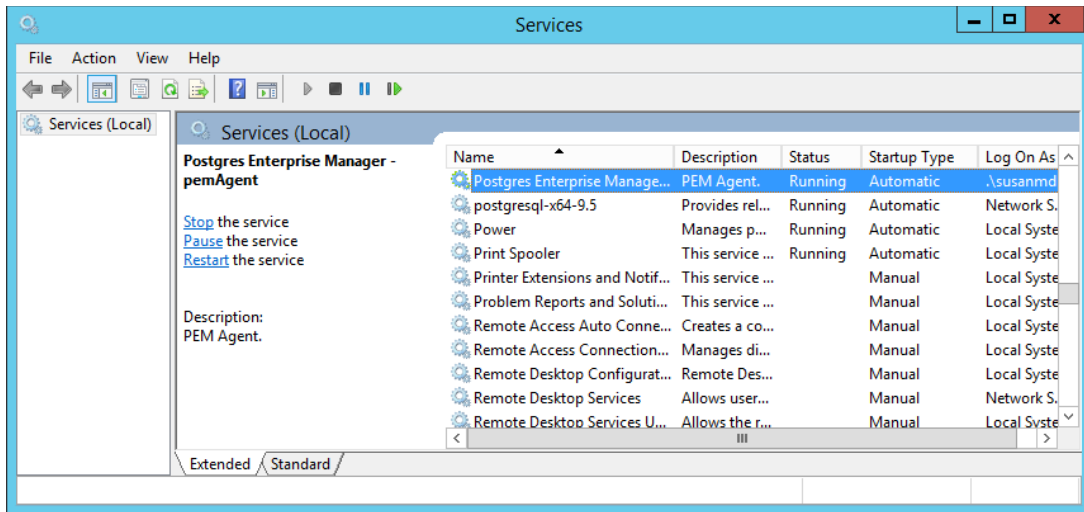


Figure 4.11 - Restarting the PEM Agent's service.

After moving the server, change the connection properties in any installed PEM clients to connect to the new host of the PEM server, agents, and monitored servers.

5 Upgrading a Graphical Installation with an RPM Package on a Linux Host

In future releases of PEM, graphical installers will not be available for Linux hosts; you will be required to use an RPM package to install or update a linux installation. The following sections walk you through the process of upgrading a PEM installation that was performed via a graphical installer on a Linux host.

5.1 Upgrading a PEM Server that was Installed with a Graphical Installer

The default installation location for the PEM server when installed by the graphical installer is `/opt/edb/pem`. In the example that follows, substitute your server installation location for `PEM_installation_path`.

1. Logout from PEM.

2. Stop the PEMHTTPD service:

```
systemctl stop PEMHTTPD
```

3. Set SELinux to permissive mode and install the `epel` packages:

```
setenforce 0
```

```
yum install epel-release*
```

4. Install the EDB repository configuration file:

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

5. When the repository configuration file installation completes, use your choice of editor to modify the `dependencies` and `tools` repository definitions, ensuring that the repository definitions are enabled and providing the correct repository credentials. For example, to use `vi`, enter:

```
vi /etc/yum.repos.d/edb.repo
```

6. The `yum makecache` command downloads the metadata for the currently enabled repositories; when the command completes, check the available packages to confirm that the list includes the latest PEM server:

```
yum makecache

yum list edb-pem-server
```

7. Install the PEM server RPM; when the installation completes, use the `yum info` command to confirm the installation details:

```
yum install edb-pem-server

yum info edb-pem-server
```

8. After installation, copy the `agent.cfg` file from the current location (the location required by the graphical installer) to the location required by the RPM package:

```
cp /PEM_installation_path/agent/etc/agent.cfg
/usr/edb/pem/agent/etc/agent.cfg
```

9. Use your choice of editor to open the agent configuration file; for example, `vi`:

```
vi /usr/edb/pem/agent/etc/agent.cfg
```

Then, set the value of the `ca_file` parameter:

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

10. Copy the `pem.db` file (and other required files) to the RPM installation location and change the file ownership:

```
cp -r /PEM_installation_path/server/share/pemhome/.pem/*
/var/lib/pemhome/.pem/
```

```
chown -R pem:pem /var/lib/pemhome/.pem/
```

11. Change the home directory in the `passwd` file from the location identified by the graphical installer to the RPM location as follows:

```
usermod -m -d /var/lib/pemhome pem
```

```
cat /etc/passwd | grep pem
```

12. Take a backup of the PEM service file and agent certificates:

```
cp /usr/lib/systemd/system/pemagent.service  
/usr/lib/systemd/system/pemagent.service_bkp
```

```
mv /root/.pem/agent1.key /root/.pem/agent1.key.bkp
```

```
mv /root/.pem/agent1.crt /root/.pem/agent1.crt.bkp
```

13. Uninstall the PEM server using the graphical uninstaller:

```
/PEM_installation_path /server/uninstall-pemserver
```

14. Execute the PEM RPM configuration script; when prompted, provide the backend database details: the script should run without generating errors:

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

15. Restore the service file backup and agent certificates to original location.

```
cp /usr/lib/systemd/system/pemagent.service_bkp  
/usr/lib/systemd/system/pemagent.service
```

```
mv /root/.pem/agent1.crt.bkp /root/.pem/agent1.crt
```

```
mv /root/.pem/agent1.key.bkp /root/.pem/agent1.key
```

16. Enable the pemagent service and start the pemagent and httpd services.

```
systemctl enable pemagent
```

```
systemctl start pemagent
```

```
systemctl start httpd
```

17. Launch the PEM web interface. Check the server and agent to confirm the PEM version, server status, and schema version. At this point, everything should be up and running.

5.2 Upgrading a PEM agent Installed with a Graphical Installer

The default installation location for the PEM server when installed by the graphical installer is `/opt/edb/pem`. In the example that follows, substitute your server installation location for `PEM_installation_path`.

1. Use the version specific command to stop the `pemagent` service.

On RHEL or CentOS 7.x:

```
systemctl stop pemagent
```

On RHEL or CentOS 6.x:

```
/etc/init.d/pemagent stop (RHEL -6)
```

2. Set SELinux to permissive mode and install the supporting `epel` packages :

```
setenforce 0
```

```
yum install epel-release*
```

3. Set SELinux to permissive mode and install the `epel` packages:

```
setenforce 0
```

```
yum install epel-release*
```

4. Install the EDB repository configuration file:

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

5. When the repository configuration file installation completes, use your choice of editor to modify the `dependencies` and `tools` repository definitions, ensuring that the repository definitions are enabled and providing the correct repository credentials:

```
vi /etc/yum.repos.d/edb.repo
```

6. The `yum makecache` command downloads the metadata for the currently enabled repositories; when the command completes, check the available packages to confirm that the list includes the latest PEM agent:

```
yum makecache
```

```
yum list edb-pem-agent
```

7. Install the PEM agent RPM; when the installation completes, you can use the `yum info` command to confirm installation information for the PEM agent:

```
yum install edb-pem-agent

yum info edb-pem-agent
```

8. After installation, copy the PEM agent configuration file (`agent.cfg`) from the previous location to the location required by the RPM installer:

```
cp /PEM_installation_path/agent/etc/agent.cfg
  /usr/edb/pem/agent/etc/agent.cfg
```

9. Use your choice of editor to open the agent configuration file; for example, `vi`:

```
vi /usr/edb/pem/agent/etc/agent.cfg
```

Then, set the value of the `ca_file` parameter:

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

10. Take a backup of the service file and agent certificates.

On RHEL or CentOS 7.x, use the following command to back up the service file:

```
cp /usr/lib/systemd/system/pemagent.service
  /usr/lib/systemd/system/pemagent.service_bkp
```

On RHEL or CentOS 6.x, use the following command to back up the service file:

```
cp /etc/init.d/pemagent /etc/init.d/pemagent_bkp
```

Then, copy the agent certificates; in the following commands, `agent_id` should specify the agent identifier (for example, `agent2` or `agent3`):

```
mv /root/.pem/agent_id.key /root/.pem/agent_id.key.bkp

mv /root/.pem/agent_id.crt /root/.pem/agent_id.crt.bkp
```

11. Uninstall the PEM agent using bitrock uninstaller

```
/PEM_installation_path/agent/uninstall-pemagent
```

12. Use version specific commands to restore the service file backup and agent certificates to original location.

For example, on a RHEL or CentOS 7.x host:

```
cp /usr/lib/systemd/system/pemagent.service_bkp  
/usr/lib/systemd/system/pemagent.service
```

On a RHEL or CentOS 6.x host:

```
cp /etc/init.d/pemagent /etc/init.d/pemagent_bkp
```

Then, move the agent certificate files:

```
mv /root/.pem/agent_id.key.bkp /root/.pem/agent_id.key
```

```
mv /root/.pem/agent_id.crt .bkp /root/.pem/agent_id.crt
```

13. Enable the pemagent service, and start pemagent and httpd. On a RHEL or CentOS 7.x host, use the commands:

```
systemctl enable pemagent
```

```
systemctl start pemagent
```

On a RHEL or CentOS 6.x host:

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

At this point, the PEM agent should be up and running; you can use the PEM web interface to check the agent version and status.

6 Troubleshooting

6.1 The pem.alert Table Fails to Restore

When restoring the pem backing database from backup, you may encounter an error during the restoration of the pem.alert table. This is caused by a missing table pre-requisite for the table - the pg_restore utility may restore the pem.alert pre-requisites *after* it attempts to restore pem.alert.

If this happens, the output from pg_restore will include error messages that refer to the alert table:

```
pg_restore: [archiver (db)] could not execute query: ERROR:
insert or update on table "alert_history" violates foreign key
constraint "alert_history_alert_id_fkey"
DETAIL:  Key (alert_id)=(3) is not present in table "alert".
    Command was: ALTER TABLE ONLY alert_history
    ADD CONSTRAINT alert_history_alert_id_fkey FOREIGN KEY
(alert_id) REFERENCES alert(id) ON...
pg_restore: creating FK CONSTRAINT alert_status_alert_id_fkey
pg_restore: [archiver (db)] Error from TOC entry 3265; 2606 18355
FK CONSTRAINT alert_status_alert_id_fkey postgres
pg_restore: [archiver (db)] could not execute query: ERROR:
insert or update on table "alert_status" violates foreign key
constraint "alert_status_alert_id_fkey"
DETAIL:  Key (alert_id)=(1) is not present in table "alert".
    Command was: ALTER TABLE ONLY alert_status
    ADD CONSTRAINT alert_status_alert_id_fkey FOREIGN KEY
(alert_id) REFERENCES alert(id) ON U...
```

If you encounter this problem, restore the pem database before restoring the pem.alert table. Restoring the pem database will install the pre-requisites for pem.alert, and the restoration of the table should complete as expected.

7 Uninstalling Postgres Enterprise Manager™

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

Component	Uninstaller name	Default location
PEM server	uninstall-pemserver	/opt/edb/pem/server
PEM client	uninstall-pemclient	/opt/edb/pem/client
PEM agent	uninstall-pemagent	/opt/edb/pem/agent
SQL Profiler	uninstall-sqlprofiler	/opt/PostgreSQL/x.x Where <i>x.x</i> specifies the Postgres version.

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.