Postgres Enterprise Manager

Release 7.14

Upgrade and Migration Guide

Jun 18, 2020

Contents

This guide provides detailed information about upgrading the Postgres Enterprise Manager (PEM) Components:

- **Upgrading a PEM Installation** - This section provides information about upgrading your PEM server, PEM Agent and SQL Profiler from one major version to another (i.e. from 6.0 to 7.14).

- **Upgrading the Backing Database** - This section provides detailed information about upgrading the backing database, while maintaining the same version of the PEM Server.

- **Moving a PEM Server** –This section provides detailed information about moving the PEM server from one host to another host.

- **Troubleshooting** –This section provides detailed information about troubleshooting the errors that you may encounter during PEM upgrade.

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

Upgrading a PEM Installation

The process of upgrading a PEM installation is platform-specific. You can update a PEM agent or server on a Windows host by using the PEM graphical installer available for Windows. Prior to PEM 7.8 release, PEM agent or server could be installed on Linux either by using the graphical installer or by using the RPMs. From PEM version 7.8 onwards, PEM graphical installers for Linux are being discontinued. To update a PEM agent or server on a Linux host from any lower version to PEM 7.9 or higher versions, you must use RPMs.

Links to PEM and SQL Profiler installers and RPMs are available at the EnterpriseDB website.

1.1 Upgrading a PEM Installation on Windows host

To upgrade PEM component software on Windows hosts, simply invoke a newer version of the PEM component installers in the following order: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

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 agent and server, you should upgrade SQL Profiler; this step is platform-specific.

The following sections will walk you through the upgrade process on a Windows host, step-by-step.

1.1.1 Upgrading a PEM Agent on Windows host

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.

You can invoke the installer by right-clicking on the downloaded installer's icon, and selecting `Run as Administrator`. The `PEM Agent Setup Wizard` opens, welcoming you.

Fig. 1.1: *The PEM Agent installer welcome window*

Read and accept the `License Agreement` before clicking `Next` to continue.

Fig. 1.2: *The PEM license agreement*

The setup wizard will automatically detect an existing agent, and upgrade the installed version. Click `Next` to continue.
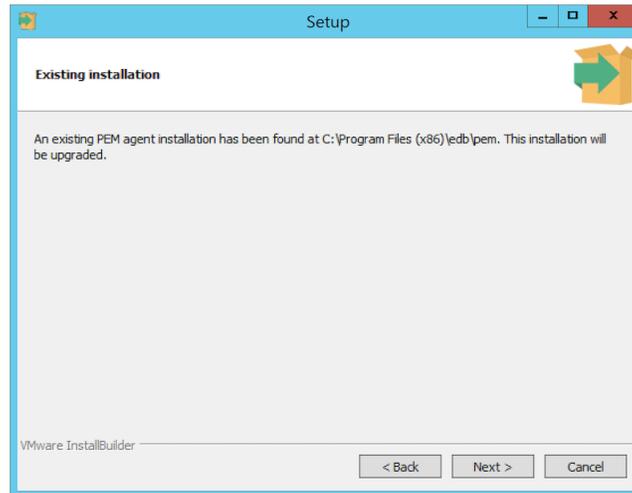
Fig. 1.3: *The installer detects an existing installation*

The `pemAgent service account` dialog may prompt you for the password of the account under which the PEM agent service runs.
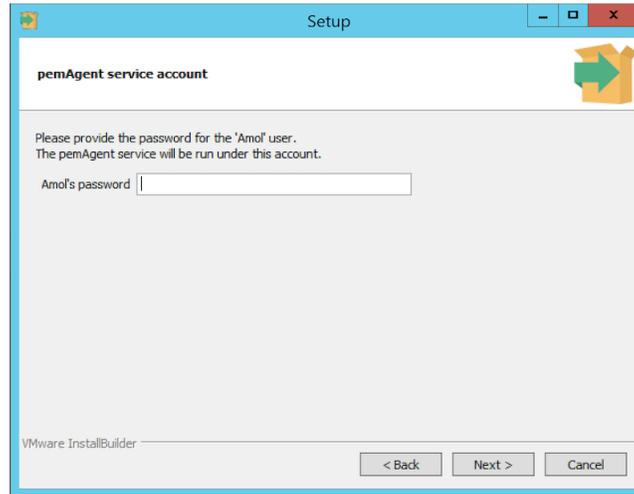
Fig. 1.4: *pemAgent Service Account password*

If prompted, provide the password, and click `Next` to continue.

When the `Ready to Install` dialog informs you that the installation is about to begin, click `Next` to continue.
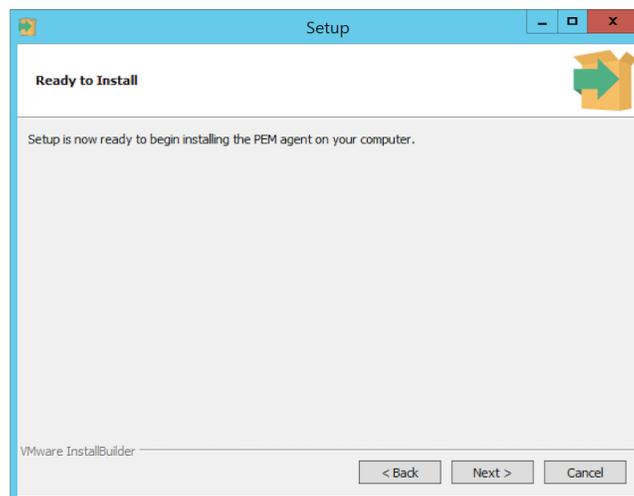
Fig. 1.5: *The PEM Agent will perform the upgrade*

The setup wizard displays progress bars to inform you of each component that is being installed.

The `PEM Agent Setup Wizard` will inform you when the installation completes. Click `Finish` to exit the wizard and close the window.
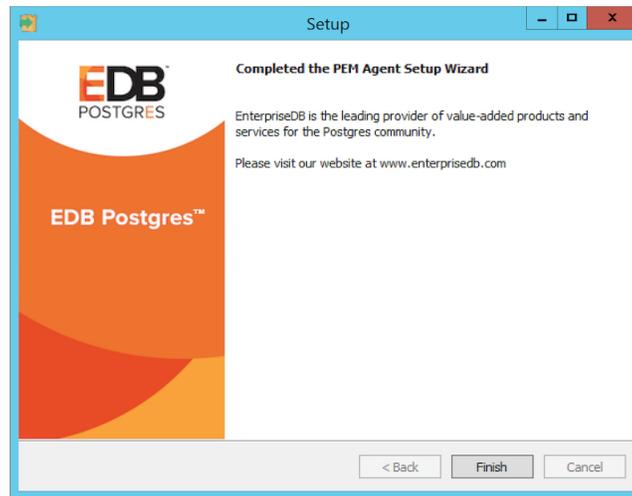
Fig. 1.6: *The PEM Agent Setup Wizard has finished the update*

After the installation completes, a window pops-up to restart the machine. Click `yes` to restart the machine and the httpd services.
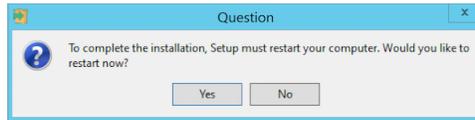
Fig. 1.7: *The PEM Agent Wizard - Restart pop-up*

Note: If you have already configured or are planning to configure any shell/batch script run by a Windows agent that is upgraded from any lower version to version 7.11, you must set the `AllowBatchJobSteps` parameter to `True` in the `agent.cfg` file. By default, the PEM agent will not execute any batch/shell script.

1.1.2 Upgrading the PEM Server on a Windows host

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).

You can invoke the installer by right-clicking on the downloaded installer's icon, and selecting `Run as Administrator`.

Fig. 1.8: *The PEM Server Setup Wizard*

The `PEM Server Setup Wizard` welcomes you, as shown in the image. Click `Next` to continue to the `License Agreement`.

The `PEM server setup wizard` will prompt you to accept the `License Agreement`. 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`.
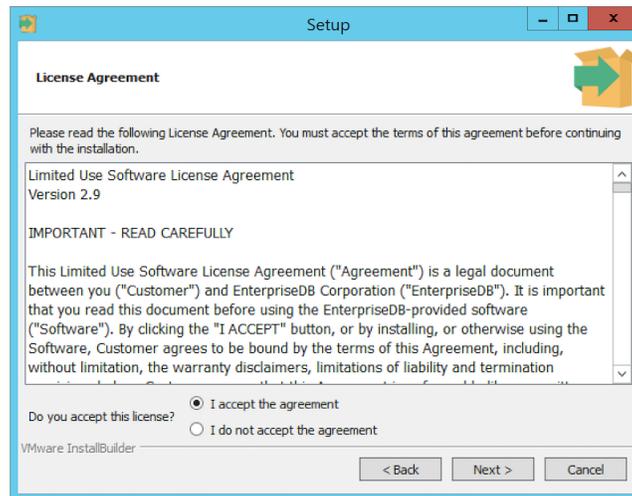
Fig. 1.9: *Accept the License Agreement to continue*

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. Click `Next` to continue.
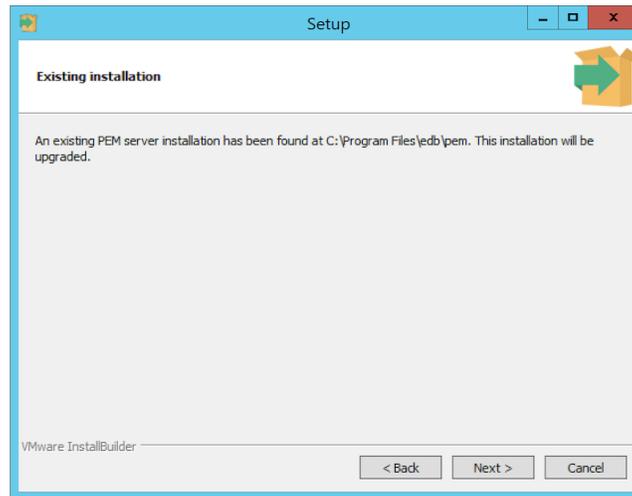
Fig. 1.10: *The PEM server installer detects an existing PEM server Installation*

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.
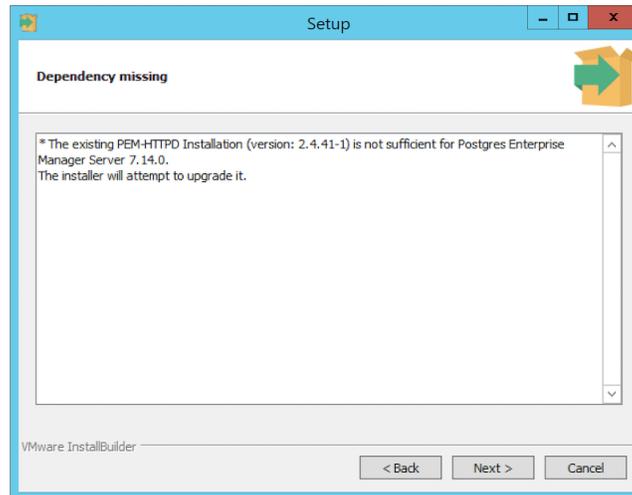
Fig. 1.11: *The installation requires dependency upgrades*

When the installation wizards complete the dependency upgrades, then a window pops-up asking whether you want to restart the machine or not.
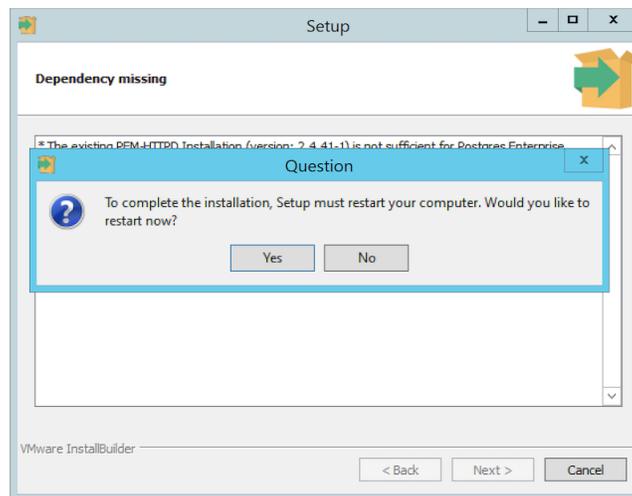
Fig. 1.12: *PEM Server Upgrade - Machine restart pop-up*

Click on `No` to continue the upgrade process.

The wizard then opens the `Database Server Installation Details` dialog, prompting you for connection credentials for database superuser of the PEM backing database. 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.
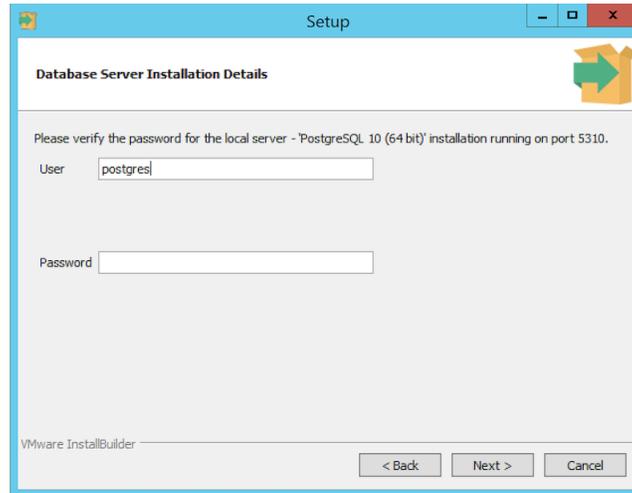
Fig. 1.13: *Provide connection information for the backing database*

The `pemAgent service account` dialog may prompt you for the password of the account under which the PEM agent service runs.
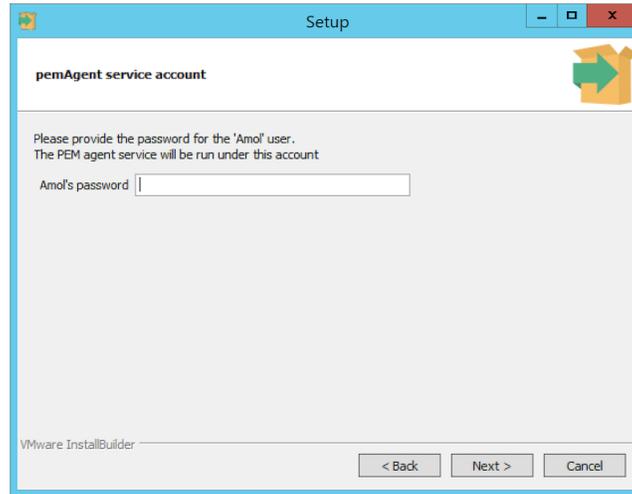
Fig. 1.14: *pemAgent Service Account password*

If prompted, provide the password, and click `Next` to continue.
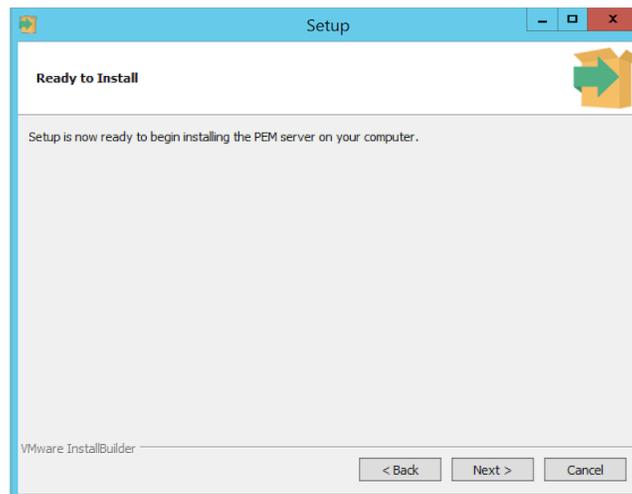
Fig. 1.15: *The PEM Setup Wizard is ready to install the PEM server*

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

During the installation, progress bars will keep you informed of the progress of the update.
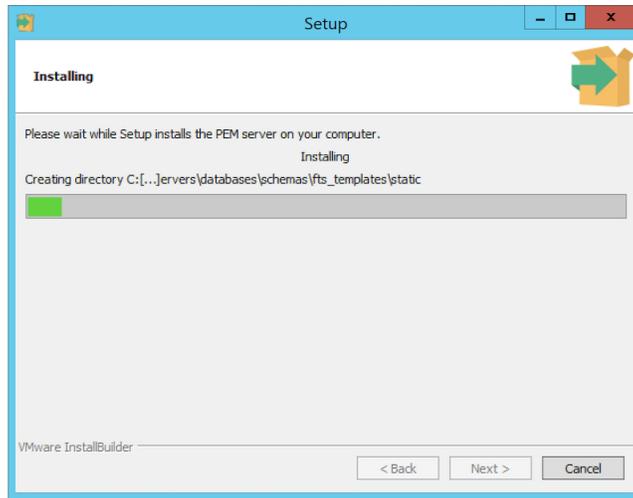
Fig. 1.16: *The PEM Server installation in progress*

After updating the PEM server (and the agent that resides on the same host as the PEM server) and configuring the web service, the PEM setup wizard notifies you of the port on which the service is listening. Use this port number when connecting to the PEM Server with the PEM client.
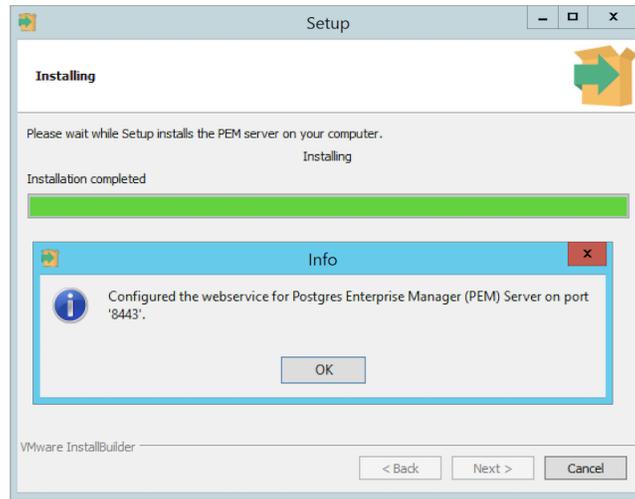
Fig. 1.17: *The setup wizard configures the PEM web service*

Click `OK` to close the `Info` popup. The PEM server setup wizard informs you that the installation is complete.
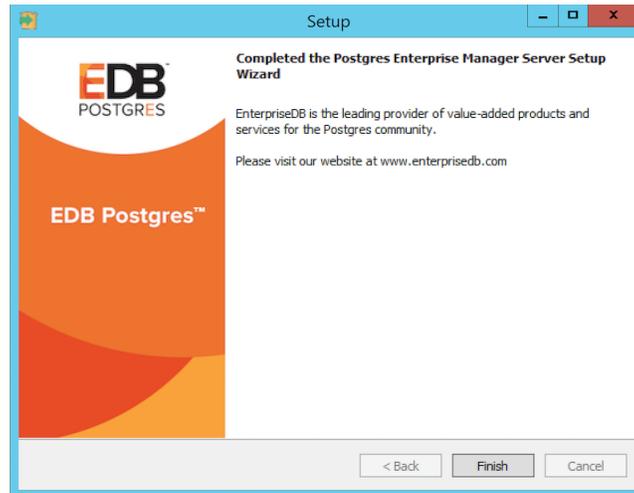
Fig. 1.18: *The PEM Server upgrade is complete*

If the window pops-up asking to restart the machine, then click on `yes` to restart the machine and hence the httpd service.

If you have installed the PEM backend database server and PEM-HTTPD on different hosts, then you must run the PEM server installer twice – once on each host. Extract the language pack installer, and install it on the host of PEM-HTTPD before invoking the PEM installer. Include the following keywords when invoking the installer to extract the language pack:

```
--extract-languagepack <path>
```

Where `<path>` specifies an existing path for extracting the language pack installer.

Note: By default EDB Languagepack is installed in `C:\edb\languagepack\v1`.

If you are upgrading the PEM Server via StackBuilder Plus then you might face the error shown below; after displaying the error, PEM will say that installation is completed. Please note that the installation is not done and you will need to do the installation by invoking the installer file from the location where it is downloaded.
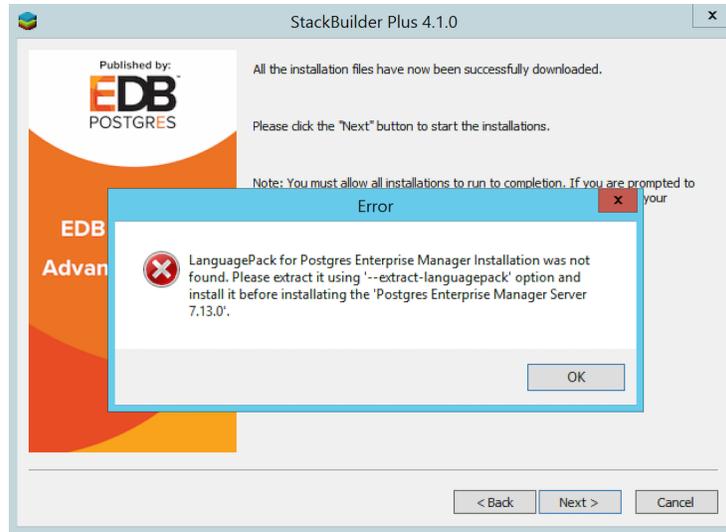
Fig. 1.19: *The PEM Server upgrade error*

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 *Upgrading the Backing Postgres Database*.

1.1.3 Upgrading SQL Profiler on 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, on a Windows host the script resides in the `share\contrib` directory under your Advanced Server or PostgreSQL installation.

   You can use the following server-specific commands. For PostgreSQL:

   ```
   psql -f C:\Program Files\PostgreSQL\<x>\share\contrib\uninstall-sql-profiler.sql -
   ↪d postgres -U postgres
   ```

   Where, `x` is the version of PostgreSQL and `-d` specifies the name of the maintenance database.

   For Advanced Server:

   ```
   psql -f C:\Program Files\edb\as<x>\share\contrib\uninstall-sql-profiler.sql -d
   ↪edb -U enterprisedb
   ```

   Where `x` is the version of Advanced Server and `-d` specifies the name of the maintenance database.

2. Invoke the new SQL Profiler installer on each node you wish to profile; run the installer as `Administrator`.

   For PostgreSQL:

   ```
   sqlprofiler-pg-<x>-<y>-windows-x64.exe
   ```

   Where, `x` is the version of the PostgreSQL and `y` is the version of SQL Profiler. For example: `sqlprofiler-pg-12-7.14.0-1-windows-x64.exe`.

   For Advanced Server:

   ```
   sqlprofiler-edb-as<x>-<y>-windows-x64.exe
   ```

   Where `x` is the version of Advanced Server and `y` is the version of SQL Profiler. For example: `sqlprofiler-edb-as12-7.14.0-1-windows-x64.exe`.

   The SQL Profiler installer will detect the existing `SQL Profiler` installation and upgrade it with the latest version of SQL Profiler.

3. Run the `sql-profiler.sql` script file in the maintenance database.

   For PostgreSQL:

   ```
   psql -f C:\Program Files\PostgreSQL\<x>\share\contrib\sql-profiler.sql -d
   ↪postgres -U postgres
   ```

   Where `x` is the version of PostgreSQL and `-d` specifies the name of the maintenance database.

   For Advanced Server:

   ```
   psql -f C:\Program Files\edb\as<x>\share\contrib\sql-profiler.sql -d edb -U
   ↪enterprisedb
   ```

Where, `x` is the version of Advanced Server and `-d` specifies the name of the maintenance database.

4. Then, restart the Postgres Server to resume profiling the node from a PEM client.

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

1.2 Upgrading a PEM Installation with an RPM Package on a Linux Host

The following sections will walk you through the upgrade process on a Linux host, step-by-step.

Prerequisites:

If you are using a version of Postgres or Advanced Server prior to version 10, before the upgrade you must install the `libs` package for version 10 or above on the system where the PEM server is installed. Use the following commands to install the `libs` version 10 or above:

- **For Advanced Server:** `yum install edb-as<version>-server-libs`
- **For Postgres:** `yum install postgresql<version>-libs`

Where *version* is the Postgres or Advanced Server version whose `libs` package you want to install.

1.2.1 Upgrading a PEM Agent that was Installed with RPM Packages

You can use an RPM package to upgrade existing agents that were initially installed with 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, use the following command:

```
yum upgrade edb-pem-agent
```

If any lower version of the PEM Agent on CentOS 6.x that is registered with a PEM server is upgraded to 7.10, you may see the following warning:

```
warning:  %postun(edb-pem-agent-7.9.0-3.rhel6.x86_64) scriptlet failed, exit
status 1
```

```
Non-fatal POSTUN scriptlet failure in rpm package edb-pem-agent
```

In spite of the above warning, the upgrade will successfully complete and the agent is automatically restarted after upgrade.

Note: If you have already configured or planning to configure any shell/batch script run by a Linux agent that is upgraded from any lower version to 7.11 or later version, the user for the `batch_script_user` parameter must be specified in the `agent.cfg` file. It is strongly recommended that a non-root user is used to run the scripts. Using the root user may result in compromising the data security and operating system security. However, if you want to restore the pemagent to its original settings using root user to run the scripts, then the `batch_script_user` parameter value must be set to `root`.

1.2.2 Upgrading a PEM Server that was Installed with RPM Packages

If you have installed your PEM server with an RPM package, you can use an RPM package to upgrade your PEM server.

If you wish to upgrade PEM server that is installed on a machine in isolated network, you need to create PEM repository on that machine before you upgrade the PEM server. For more information on creating PEM repository in isolated network, see *Creating a PEM Repository on an Isolated Network*.

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
   ```

2. 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 /root/.pem.
-ci	CIDR formatted network address range that agents will connect to the server from, to be added to the server's pg_hba.conf file. For example, 192.168.1.0/24. The default is 0.0.0.0/0.
-dbi	The directory for the database server installation. For example, /usr/edb/as10 for Advanced Server or /usr/pgsql-10 for PostgreSQL.
-ds	The unit file name of the PEM database server. For Advanced Server, the default file name is edb-as-10; for PostgreSQL, it is postgresql-10.
-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 pemagent.
-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.
-un	Uninstalls PEM server.
-h	Displays help.

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
```

If you are upgrading a 7.10 PEM Server on RHEL 6.x to PEM version 7.11, you may see the following error text during the upgrade:

```
/var/tmp/rpm-tmp.<xxxxxx>:  line 9:  Requires::  command not found
```

In spite of the above error, the upgrade will be completed successfully.

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 or PEM agent, please see the *PEM Installation Guide*, available at:

https://www.enterprisedb.com/edb-docs/p/edb-postgres-enterprise-manager

---

1.2.3 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, on a Linux host the script resides in the `share/contrib` directory under your Advanced Server or PostgreSQL installation.

   You can use the following server-specific command:

   For PostgreSQL:

   ```
   /usr/pgsql-<x>/bin/psql -f /usr/pgsql-<x>/share/contrib/uninstall-sql-profiler.
   ↪sql -d postgres -U postgres
   ```

   Where, `x` is the version of PostgreSQL and `-d` specifies the name of the maintenance database.

   For Advanced Server:

   ```
   /usr/edb/as<x>/bin/psql -f /usr/edb/as<x>/share/contrib/uninstall-sql-
   ↪profiler.sql -d edb -U enterprisedb
   ```

   Where, `x` is the version of Advanced Server and `-d` specifies the name of the maintenance database.

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

   For PostgreSQL:

   ```
   yum install postgresql<x>-sqlprofiler
   ```

   Where, `x` is the version of the PostgreSQL.

   For Advanced Server:

   ```
   yum install edb-as<x>-server-sqlprofiler
   ```

   Where `x` is the version of Advanced Server.

The installer will detect the existing `SQL Profiler` installation and upgrade with the latest version of SQL Profiler.

Please see the following example of upgrading SQL Profiler for PostgreSQL:

```
[root@localhost Downloads]# yum install postgresql12-sqlprofiler
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirrors.piconets.webwerks.in
 * epel: hkg.mirror.rackspace.com
 * extras: mirrors.piconets.webwerks.in
 * updates: mirrors.piconets.webwerks.in
Resolving Dependencies
--> Running transaction check
---> Package postgresql12-sqlprofiler.x86_64 0:7.12.0-1.rhel7 will be updated
---> Package postgresql12-sqlprofiler.x86_64 0:7.14.0-1.rhel7 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package                  Arch           Version              Repository    Size
================================================================================
Updating:
 postgresql12-sqlprofiler x86_64         7.14.0-1.rhel7       edb           88 k

Transaction Summary
================================================================================
Upgrade  1 Package

Total download size: 88 k
Is this ok [y/d/N]: y
Downloading packages:
No Presto metadata available for edb
postgresql12-sqlprofiler-7.14.0-1.rhel7.x86_64.rpm                | 88 kB  00:00:04
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Updating   : postgresql12-sqlprofiler-7.14.0-1.rhel7.x86_64                1/2
  Cleanup    : postgresql12-sqlprofiler-7.12.0-1.rhel7.x86_64                2/2
  Verifying  : postgresql12-sqlprofiler-7.14.0-1.rhel7.x86_64                1/2
  Verifying  : postgresql12-sqlprofiler-7.12.0-1.rhel7.x86_64                2/2

Updated:
  postgresql12-sqlprofiler.x86_64 0:7.14.0-1.rhel7

Complete!
```

Fig. 1.20: *Upgrading SQL Profiler*

3. Then, run the `sql-profiler.sql` script file in the maintenance database.

   For PostgreSQL:

   ```
   /usr/pgsql-<x>/bin/psql -f /usr/pgsql-<x>/share/contrib/sql-profiler.sql -d␣
   →postgres -U postgres
   ```

   Where, `x` is the version of PostgreSQL and `-d` specifies the name of the maintenance database.

   For Advanced Server:

   ```
   /usr/edb/as<x>/bin/psql -f /usr/edb/as<x>/share/contrib/sql-profiler.sql -d␣
   →edb -U enterprisedb
   ```

   Where, `x` is the version of Advanced Server and `-d` specifies the name of the maintenance database.

4. Restart the PostgreSQL/Advanced Server to resume profiling the node from the PEM Client.

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

1.3 Creating a PEM Repository on an Isolated Network

You can create a local repository to act as a host for the PEM RPM packages if the server on which you wish to upgrade PEM cannot directly access the EnterpriseDB repository. Please note that this is a high-level overview of the steps required; you may need to modify the process for your individual network. To create and use a local repository, you must:

1. Use the following commands on a system with Internet access to download the dependencies for PEM:

```
yum install yum-plugin-downloadonly

mkdir /tmp/<pem_dir>

yum install --downloadonly --downloaddir=/tmp/<pem_dir>/ edb-pem

mkdir /tmp/<epel_dir>

yum install --downloadonly --downloaddir=/tmp/<epel_dir>/ epel-release*
```

Where `<pem_dir>` and `<epel_dir>` are the local directories that you create for downloading the RPMs.

2. Copy the directories `/tmp/<pem_dir>` and `/tmp/<epel_dir>` to the machine that is in the isolated network.

3. Create the repositories:

```
createrepo /tmp/<pem_dir>

createrepo /tmp/<epel_dir>
```

4. Create a repository configuration file called `/etc/yum.repos.d/pem.repo` with connection information that specifies:

```
[pemrepo]
name=PEM Repository
baseurl=file:///tmp/pem7/
enabled=1
gpgcheck=0
```

5. Create a repository configuration file called `/etc/yum.repos.d/epel.repo` with connection information that specifies:

```
[epelrepo]
name=epel Repository
baseurl=file:///tmp/pem7/
enabled=1
gpgcheck=0
```

After specifying the location and connection information for your local repository, you can use yum commands to install or upgrade PEM server:

- To install PEM server:

  ```
  yum install edb-pem
  ```

- To upgrade PEM server:

  ```
  yum upgrade edb-pem
  ```

For more information about creating a local yum repository, visit: https://wiki.centos.org/HowTos/CreateLocalRepos

1.4 Upgrading a Graphical PEM Installation on a Linux Host

The following section walks you through the process of upgrading a PEM installation that was performed via a graphical installer on a Linux host

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

Prerequisites:

If you are using a version of Postgres or Advanced Server released prior to version 10, then before the upgrade you need to install the `libs` package for version 10 or above on the system where the PEM server is installed. Use the following commands to install `libs` version 10 or above:

- **For Advanced Server:** `yum install edb-as<version>-server-libs`
- **For Postgres:** `yum install postgresql<version>-libs`

Where *version* is the Postgres or Advanced Server version whose libs package you want to install.

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 on PEM server. In case PEM server and web server are on two different systems, run the command on the web server:

   `systemctl stop PEMHTTPD`

3. Install the `epel-release` package by running any one of the following commands:

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

- `yum install epel-release`

   Note: You may need to enable the `[extras]` repository definition in the `CentOS-Base.repo` file (located in `/etc/yum.repos.d`).
   ---

4. You must also have credentials that allow access to the EnterpriseDB repository. For information about requesting credentials, visit:

   https://info.enterprisedb.com/rs/069-ALB-339/images/Repository%20Access%2004-09-2019.pdf

5. After receiving your repository credentials you can:

   (a) Create the repository configuration file.

   (b) Modify the file, providing your user name and password.

   Creating a Repository Configuration File

   To create the repository configuration file, assume superuser privileges, and invoke the following command:

   ```
   yum -y install https://yum.enterprisedb.com/edb-repo-rpms/
   edb-repo-latest.noarch.rpm
   ```

   The repository configuration file is named `edb.repo`. The file resides in `/etc/yum.repos.d`.

   Modifying the file, providing your user name and password

After creating the `edb.repo` file, use your choice of editor to ensure that the value of the enabled parameter is `1`, and replace the `username` and `password` placeholders in the `baseurl` specification with the name and password of a registered EnterpriseDB user.

```
[edb]
name=EnterpriseDB RPMs $releasever - $basearch
baseurl=https://<username>:<password>@yum.enterprisedb.com/edb/redhat/rhel-
↪$releasever-$basearch
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/ENTERPRISEDB-GPG-KEY
```

6. Use the `yum makecache` command to download 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
   ```

7. Install the PEM server RPM on PEM server. In case PEM server and web server are on two different systems, run the command on PEM server as well as web server. When the installation completes, use the `yum info` command to confirm the installation details:

   ```
   yum install edb-pem
   ```

   ```
   yum info edb-pem
   ```

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. Open the agent configuration file located at:

   ```
   /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. In case PEM server and web server are on two different systems, run the below commands on the web server:

    ```
    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. In case PEM server and web server are on two different systems, run the commands on PEM server as well as web server:

    ```
    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 from PEM server and web server machines:

    ```
    /<PEM_installation_path> /server/uninstall-pemserver
    ```

14. Execute the PEM RPM configuration script on PEM server and web server; when prompted, provide the backing 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 on PEM server.

    ```
    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 on PEM server and web server. Start the `pemagent` and `httpd` services on the web server.

    ```
    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. You can now uninstall PEMHTTD service from web server as it is no longer in use.

1.4.2 Upgrading a PEM Agent 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. Use the version specific command to stop the `pemagent` service.

   - On a RHEL or CentOS 7.x 0r 8.x host:

     ```
     systemctl stop pemagent
     ```

   - On a RHEL or CentOS 6.x host:

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

2. Install the supporting `epel` packages:

   ```
   yum install epel-release
   ```

3. Install the EDB repository configuration file:

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

4. When the repository configuration file installation completes, modify the `edb` repository definition, ensuring that the repository is enabled and providing your EnterpriseDB credentials. The location of the file is:

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

5. 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
   ```

6. 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
   ```

7. 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
   ```

8. Open the agent configuration file located at:

   ```
   /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
   ```

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

   - On RHEL or CentOS 7.x or 8.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_bkp /etc/init.d/pemagent
```

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
```

10. Uninstall the PEM agent using bitrock uninstaller:

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

11. 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 or 8.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; in the following commands, `<agent_id>` should specify the agent identifier (for example, agent2 or agent3):

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

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

12. Enable the `pemagent` service, and start `pemagent` and `httpd`.

   - On a RHEL or CentOS 7.x or 8.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.

Note: If you have already configured or are planning to configure any shell/batch script run by a Linux agent that is upgraded from any version prior to version 7.11, you must modify the `agent.cfg` file and specify the user for the `batch_script_user` parameter. It is strongly recommended that you use a non-root user to run the scripts. Using the root user when running a script may result in compromising the data security and operating system security. However, if you want to restore the earlier behavior of the PEM agent and use the root user to run the scripts, then you need to set the value of `batch_script_user` parameter as `root`.

1.4.3 Upgrading SQL Profiler on a Linux host

Note: The SQL Profiler graphical installers are available for PostgreSQL and Advanced Server versions prior to 11.x (i.e. 10.x, 9.6, 9.5,..).

To upgrade a SQL Profiler installation 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 PostgreSQL on a Linux host that was installed with a graphical installer, the script resides in the `share/postgresql/contrib` directory under the PostgreSQL installation; for Advanced Server, the script resides in the `share/contrib` directory under the Advanced Server installation. You can use the following server-specific commands.

   For PostgreSQL:

   ```
   /opt/PostgreSQL/<x>/bin/psql -f /opt/PostgreSQL/<x>/share/postgresql/contrib/
   ↪uninstall-sql-profiler.sql -d postgres -U postgres
   ```

   Where `x` is the version of PostgreSQL and `-d` specifies the name of the maintenance database.

   For Advanced Server:

   ```
   /opt/edb/as<x>/bin/psql -f /opt/edb/as<x>/share/contrib/uninstall-sql-profiler.
   ↪sql -d edb -U enterprisedb
   ```

   Where, `x` is the version of Advanced Server and `-d` specifies the name of the maintenance database.

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

   For a PostgreSQL host:

   ```
   sqlprofiler-pg-<x>-<y>-linux.exe
   ```

   Where `x` is the version of the PostgreSQL and `y` is the version of SQL Profiler. For example: `sqlprofiler-pg-12-7.14.0-1-linux.exe`.

   For Advanced Server:

   ```
   sqlprofiler-edb-as<x>-<y>-linux.exe
   ```

   Where `x` is the version of Advanced Server and `y` is the version of SQL Profiler. For example: `sqlprofiler-edb-as12-7.14.0-1-linux.exe`.

   The SQL Profiler installer will detect the existing `SQL Profiler` installation and upgrade it with the latest version of SQL Profiler.

3. Run the `sql-profiler.sql` script file in the maintenance database.

   For PostgreSQL:

   ```
   /opt/PostgreSQL/<x>/bin/psql -f /opt/PostgreSQL/<x>/share/postgresql/contrib/sql-
   ↪profiler.sql -d postgres -U postgres
   ```

   Where, `x` is the version of PostgreSQL and `-d` specifies the name of the maintenance database.

   For Advanced Server:

```
  /opt/edb/as<x>/bin/psql -f /opt/edb/as<x>/share/contrib/sql-profiler.sql -
→d edb -U enterprisedb

Where, ``x`` is the version of Advanced Server and ``-d`` specifies the name␣
→of the maintenance database.
```

4. Restart the PostgreSQL/Advanced Server to resume profiling the node from the PEM Client.

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

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 *Upgrading a PEM Installation*.

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), see *Moving the Postgres Enterprise Manager™ Server*.

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:

https://www.enterprisedb.com/software-downloads-postgres

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:

https://www.enterprisedb.com/downloads/modified-gpl-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:

https://www.enterprisedb.com/downloads/modified-gpl-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 of the folllowing:

- `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:

https://www.enterprisedb.com/edb-docs

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 `vcredist_x86.exe`).

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

https://www.enterprisedb.com/downloads/modified-gpl-source-code

Copy the unpacked `sslutils` folder to the Postgres installation directory (i.e. `C:\ProgramFiles\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 GETTEXTPATH=\ <path_to_gettext>

SET OPENSSLPATH=\ <path_to_openssl>

SET PGPATH=\ <path_to_pg_installation_dir>

SET ARCH=x86

msbuild sslutils.proj /p:Configuration=Release
```

Where:

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

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

path_to_pg_installation_dir specifies the location of the Postgres installation.

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

```
SET USE_PGXS=1

SET OPENSSLPATH=C:\OpenSSL-Win32

SET GETTEXTPATH="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 or 8.x, open a command line and assume the identity of a superuser. Enter the command:

```
systemctl/<service_name> stop
```

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

On Windows, you can use the `Services` dialog 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`

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
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`.
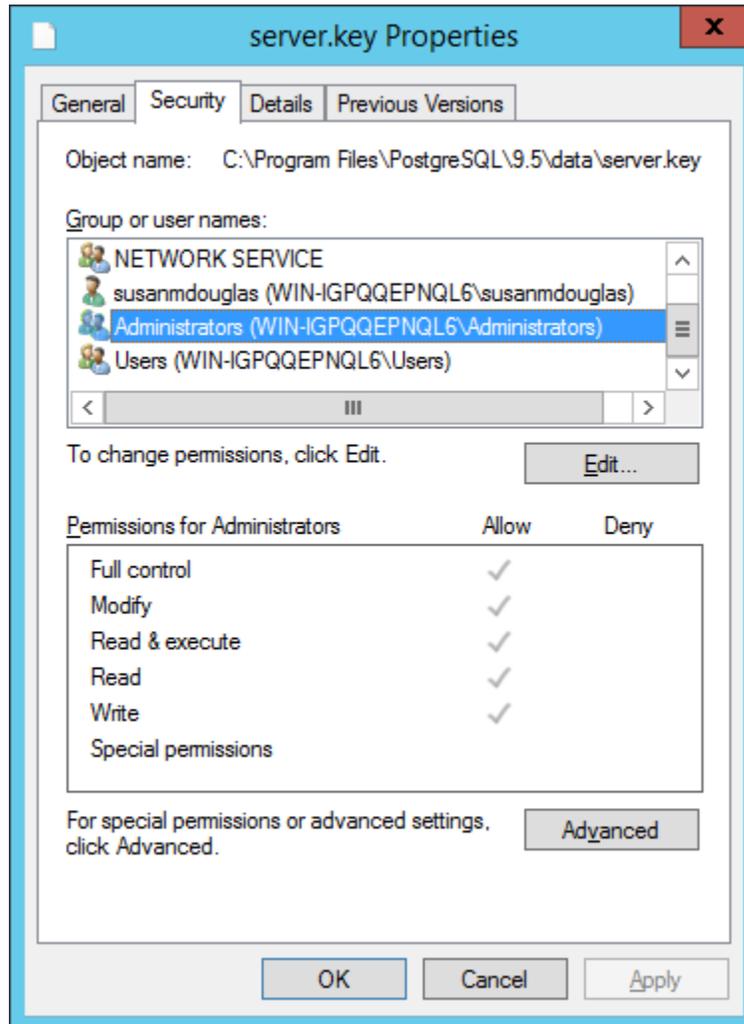

Fig. 2.1: *The Security tab.*

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 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 at the following location:

- 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/AS<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 and *x* is the minor version.

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 or 8.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.

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:

> https://www.enterprisedb.com/downloads/modified-gpl-source-code

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

> https://www.enterprisedb.com/edb-docs/p/edb-postgres-enterprise-manager

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`.
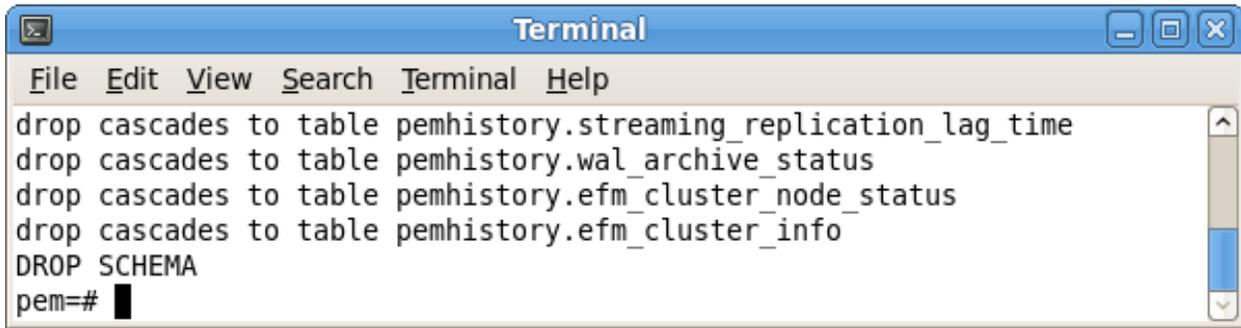
Fig. 3.1: *Dropping the pem schema*

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.

    SELECT id FROM pem.agent WHERE active = true;

You must manually create the number of agents that reside on the original PEM server, less one; the PEM server installer 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 agent <x>;

Where `<x>` specifies an agent number. Remember, `agent1` is created on the target host by the PEM server installer.

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

    GRANT pem_agent TO agent <x>;

Where `<x>` specifies an agent number.

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.

`<db_name>` specifies the name of 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.

The command shown 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.

Fig. 3.2: *Restoring a backup script*

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

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.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

Where:

    <x.x> specifies the version of PostgresSQL on your system.

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`.
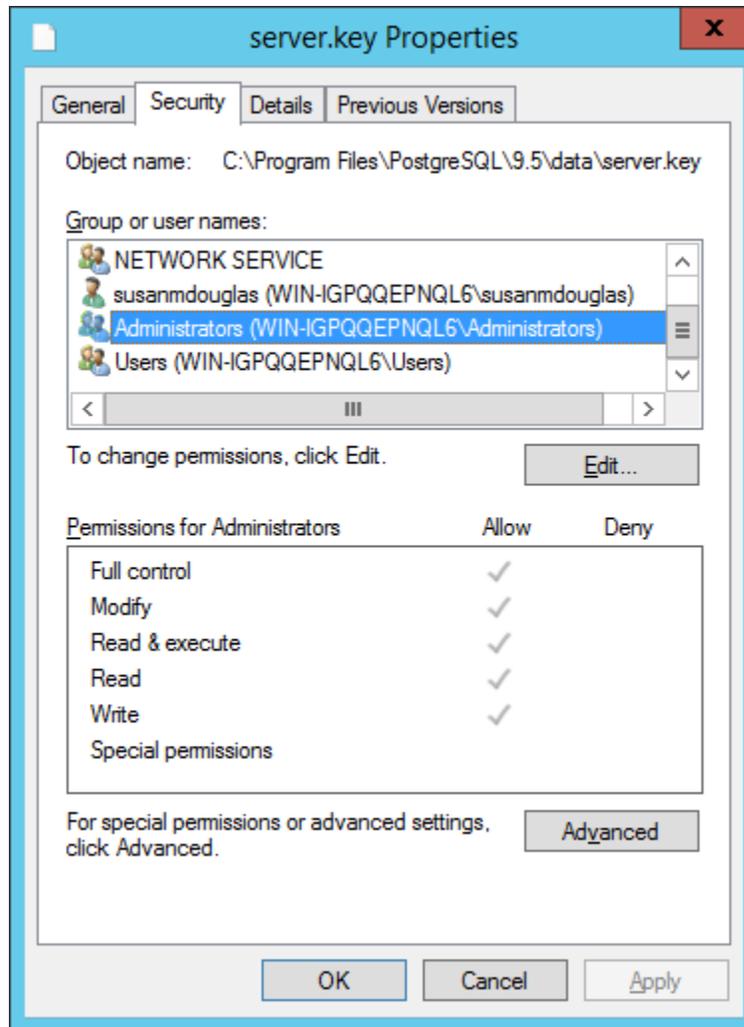
Fig. 3.3: *The Permissions tab*

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 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
agent1.key	root	-rw———-
agent1.crt	root	-rw-r–r–

If necessary, navigate to `/root/.pem`, and use the following commands to modify the permissions and ownership

of the `agent1.key` file:

```
chmod 600 agent1.key

chown root agent1.key
```

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

```
chmod 644 agent1.crt

chown root agent1.crt
```

On Windows

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

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

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

The ownership and permissions associated with the certificate files on the target machine 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, 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
```
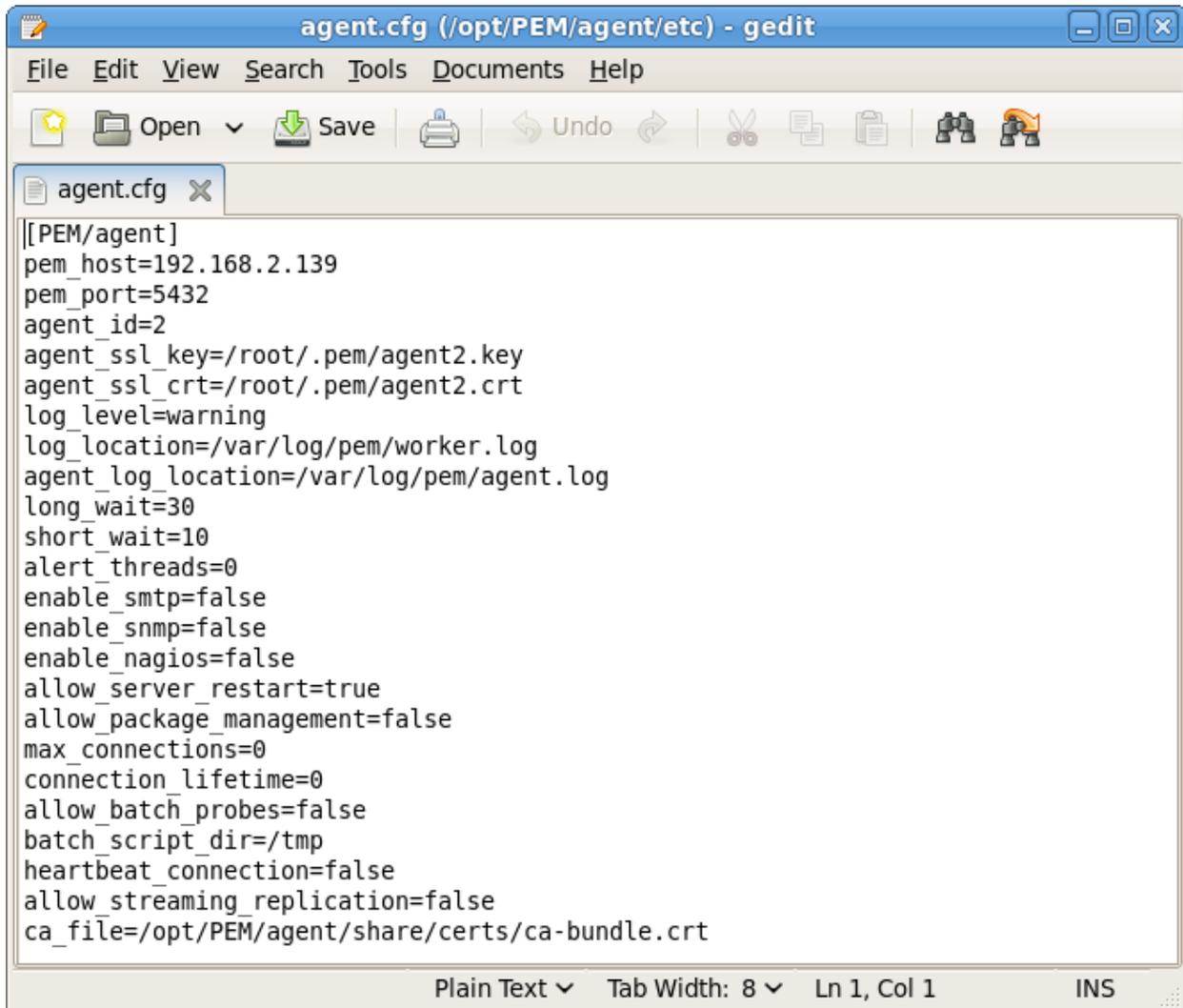

Fig. 3.4: *The agent.cfg file*

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 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
```
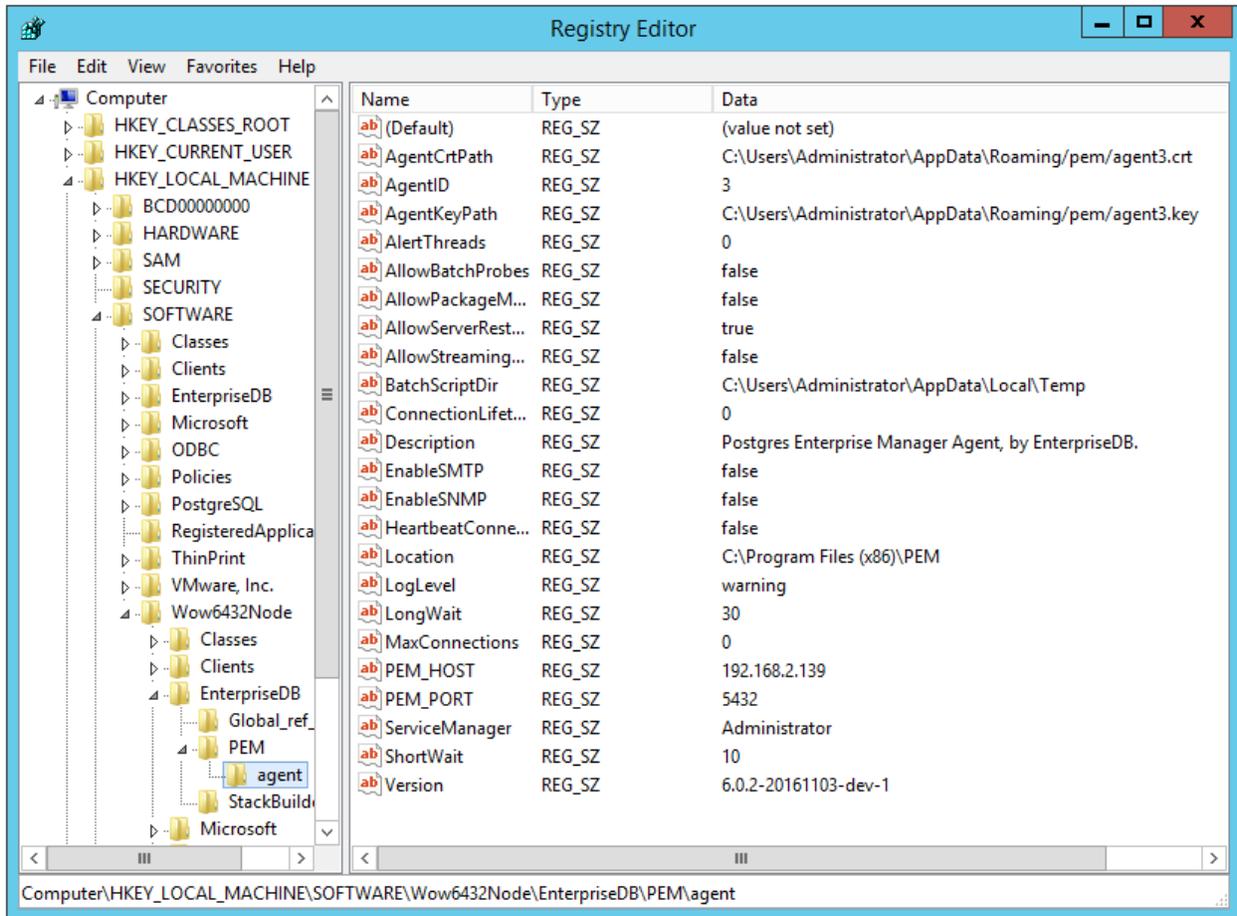
Fig. 3.5: *The Windows Registry Editor*

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.
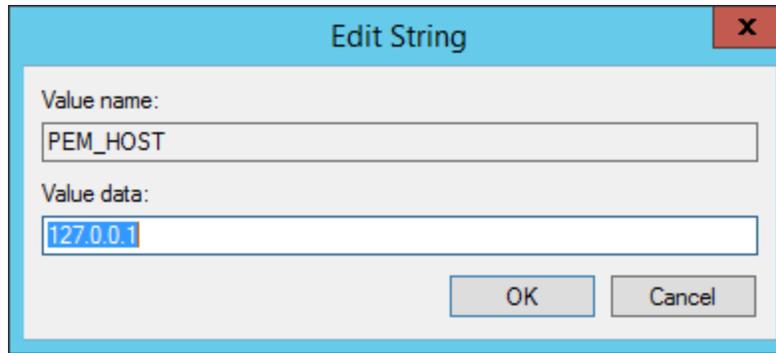
Fig. 3.6: *The Windows Registry Editor*

Use the `Edit String` dialog to make any changes to the value of the entry. 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 .
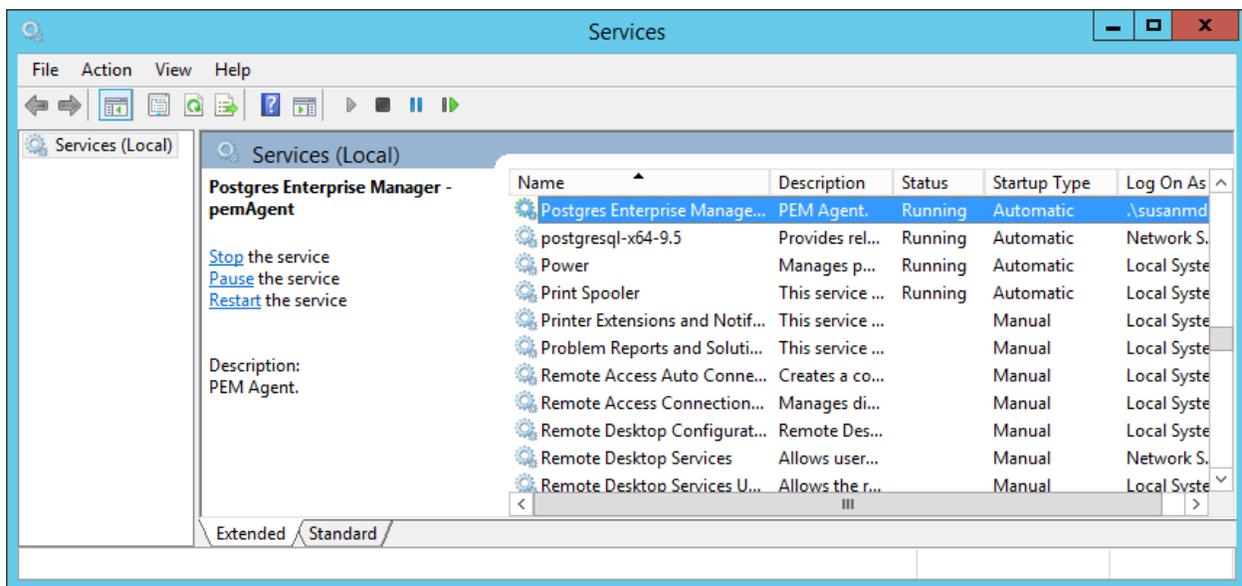

Fig. 3.7: *Restarting the PEM Agent's service*

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.

Troubleshooting

4.1 Reconfiguring the PEM Server

In certain situations, you may need to uninstall the PEM server, install it again, and reconfigure the PEM server. Use the following commands in the given sequence:

1. Use the following command to remove the PEM server configuration and uninstall:

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

2. Use the following command to remove the PEM packages:

   ```
   yum erase edb-pem-server
   ```

3. Use the following command to drop the `pem` database:

   ```
   DROP DATABASE pem
   ```

4. Move the certificates from `/root/.pem/` to another location:

   ```
   mv /root/.pem/* <new_location>
   ```

5. Move the `agent.cfg` file from `/usr/edb/pem/agent/etc/agent.cfg` to another location:

   ```
   mv /usr/edb/pem/agent/etc/agent.cfg <new_location>
   ```

6. Then, use the following command to configure the PEM server again:

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

4.2 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.

CHAPTER 5

Conclusion

EnterpriseDB® Corporation 34 Crosby Drive, Suite 201, Bedford, MA 01730, USA

T +1 781 357 3390 F +1 978 467 1307 E info@enterprisedb.com www.enterprisedb.com

- EnterpriseDB and Postgres Enterprise Manager are registered trademarks of EnterpriseDB Corporation. EDB and EDB Postgres are trademarks of EnterpriseDB Corporation. Oracle is a registered trademark of Oracle, Inc. Other trademarks may be trademarks of their respective owners.

- EDB designs, establishes coding best practices, reviews, and verifies input validation for the logon UI for EDB products where present. EDB follows the same approach for additional input components, however the nature of the product may require that it accepts freeform SQL, WMI or other strings to be entered and submitted by trusted users for which limited validation is possible. In such cases it is not possible to prevent users from entering incorrect or otherwise dangerous inputs.

- EDB reserves the right to add features to products that accept freeform SQL, WMI or other potentially dangerous inputs from authenticated, trusted users in the future, but will ensure all such features are designed and tested to ensure they provide the minimum possible risk, and where possible, require superuser or equivalent privileges.

- EDB does not that warrant that we can or will anticipate all potential threats and therefore our process cannot fully guarantee that all potential vulnerabilities have been addressed or considered.

Index