Contents

---

0 PEM Administrator’s Guide

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

For information about the platforms and versions supported by PEM, visit the EnterpriseDB website at:

https://www.enterprisedb.com/services-support/edb-supported-products-and-platforms#pem

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

1.0 PEM Administrator’s Guide

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

For information about the platforms and versions supported by PEM, visit the EnterpriseDB website at:

https://www.enterprisedb.com/services-support/edb-supported-products-and-platforms#pem

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

1.1 PEM Overview

PEM provides a number of benefits not found in any other PostgreSQL management tool:

- **Management en Masse Design**. PEM is designed for enterprise database management, and is built to tackle the management of large numbers of servers across geographical boundaries. Global dashboards keep you up to date on the up/down/performance status of all your servers in an at-a-glance fashion.
- **Distributed Architecture**. PEM is architected in a way that maximizes its ability to gather statistical information and to perform operations remotely on machines regardless of operating system platform.
- **Graphical Administration**. All aspects of database administration can be carried out in the PEM client via a graphical interface. Server startup and shutdown, configuration management, storage and security control, object creation, performance management, and more can be handled from a single console.
- **Full SQL IDE**. PEM contains a robust SQL integrated development environment (IDE) that provides ad-hoc SQL querying, stored procedure/function development, and a graphical debugger.
- **Enterprise Performance Monitoring**. PEM provides enterprise-class performance monitoring for all managed database servers. Lightweight and efficient agents monitor all aspects of each database server's operations as well as each machine's underlying operating system and provide detailed statistics back to easily navigated performance pages within the interface.
- **Proactive Alert Management**. PEM ships out-of-the-box with the ability to create performance thresholds for each key metric (e.g. memory, storage, etc.) that are monitored around-the-clock. Any threshold violation results in an alert being sent to a centralized dashboard that communicates the nature of the problem and what actions are necessary to prevent the situation from jeopardizing the overall performance of the server.
- **Simplified Capacity Planning**. All key performance-related statistics are automatically collected and retained for a specified period of time in PEM's repository. The Capacity Manager utility allows you to select various statistics and perform trend analysis over time to understand things such as peak load periods, storage consumption trends, and much more. A forecasting mechanism in the tool allows you to also forecast resource usage in the future and plan/budget accordingly.
- **Audit Manager.** The Audit Manager configures audit logging on Advanced Server instances. Activities such as connections to a database, disconnections from a database, and the SQL statements run against a database can be logged. The Audit Log dashboard can then be used to filter and view the log.
- **Log Manager.** The Log Manager wizard configures server logging parameters, with (optional) log collection into a central table. Use the wizard to specify your preference for logging behaviors such as log file rotation, log destination and error message severity. Use the Server Log dashboard to filter and review the collected server log entries.
- **SQL Workload Profiling**. PEM contains a SQL profiling utility that allows you to trace the SQL statements that are executed against one or more servers. SQL profiling can either be done in an ad-hoc or scheduled manner. Captured SQL statements can then be filtered so you can easily identify and tune poorly running SQL statements. SQL statements can also be fed into an Index Advisor on Advanced Server that analyzes each statement and makes recommendations on new indexes that should be created to help performance.
- **Expert Database Analysis**. PEM includes the Postgres Expert utility. Postgres Expert analyzes selected databases for best practice enforcement purposes. Areas such as general configuration, security setup, and much more are examined. Any deviations from recommended best practices are reported back to you, along with an explanation of each particular issue, and expert help on what to do about making things right.
- **Streaming Replication Monitoring.** You can monitor the the Streaming Replication dashboard or use options on the PEM client to promote a standby node to the master node.
- **Secure Client Connectivity.** PEM supports secure client connections through an encrypted SSH tunnel. The full-featured PEM client includes an SSH Tunnel definition dialog that allows you to provide connection information for a secure connection.
- **Wide Platform Support**. PEM supports most major Linux and Windows platforms.

1.2 Registering a Server

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

Manually Registering a Server

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

- Register your Advanced Server or PostgreSQL server with the PEM server.
- Bind the server to a PEM agent.

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

Accessing the Create – Server dialog

Note

You must ensure the `pg_hba.conf` file of the Postgres server that you are registering allows connections from the host of the PEM client before attempting to connect.

The General tab of the Create – Server dialog

Use the fields on the `General` tab to describe the general properties of the server:

- Use the `Name` field to specify a user-friendly name for the server. The name specified will identify the server in the PEM `Browser` tree control.
- You can use groups to organize your servers and agents in the tree control. Using groups can help you manage large numbers of servers more easily. For example, you may want to have a production group, a test group, or LAN specific groups. Use the `Group` drop-down listbox to select the server group in which the new server will be displayed.
- Use the `Team` field to specify a Postgres role name. Only PEM users who are members of this role, who created the server initially, or have superuser privileges on the PEM server will see this server when they logon to PEM. If this field is left blank, all PEM users will see the server.
- Use the `Background` color selector to select the color that will be displayed in the PEM tree control behind database objects that are stored on the server.
- Use the `Foreground` color selector to select the font color of labels in the PEM tree control for objects stored on the server.
- Check the box next to `Connect now?` to instruct PEM to attempt a server connection when you click the Save button. Leave `Connect now?` unchecked if you do not want the PEM client to validate the specified connection parameters until a later connection attempt.
- Provide notes about the server in the `Comments` field.

The Connection tab of the Create – Server dialog

Use fields on the `Connection tab` to specify connection details for the server:

- Specify the IP address of the server host, or the fully qualified domain name in the `Host name/address` field. On Unix based systems, the address field may be left blank to use the default PostgreSQL Unix Domain Socket on the local machine, or may be set to an alternate path containing a PostgreSQL socket. If you enter a path, the path must begin with a ”/”.
- Specify the port number of the host in the `Port` field.
- Use the `Maintenance database` field to specify the name of the initial database that PEM will connect to, and that will be expected to contain `pgAgent` schema and `adminpack` objects installed (both optional). On PostgreSQL 8.1 and above, the maintenance DB is normally called `postgres` ; on earlier versions `template1` is often used, though it is preferrable to create a `postgres` database to avoid cluttering the template database.
- Specify the name that will be used when authenticating with the server in the `Username` field.
- Provide the password associated with the specified user in the `Password` field.
- Check the box next to `Save password?` to instruct PEM to store passwords in the `~/.pgpass` file (on Linux) or `%APPDATA%\postgresql\pgpass.conf` (on Windows) for later reuse. For details, see

the `pgpass` documentation. Stored passwords will be used for all libpq based tools. To remove a password, disconnect from the server, open the server's Properties dialog and uncheck the selection.

- Use the `Role` field to specify the name of the role that is assigned the privileges that the client should use after connecting to the server. This allows you to connect as one role, and then assume the permissions of another role when the connection is established (the one you specified in this field). The connecting role must be a member of the role specified.

The SSL tab of the Create – Server dialog

Use the fields on the `SSL` tab to configure SSL:

- Use the drop-down list box in the `SSL mode` field to select the type of SSL connection the server should use. For more information about using SSL encryption, see the PostgreSQL documentation at:

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

You can use the platform-specific `File` manager dialog to upload files that support SSL encryption to the server. To access the File manager, click the icon that is located to the right of each of the following fields:

- Use the `Client certificate` field to specify the file containing the client SSL certificate. This file will replace the default `~/.postgresql/postgresql.crt` file if PEM is installed in Desktop mode, and `<STORAGE_DIR>/<USERNAME>/.postgresql/postgresql.crt` if PEM is installed in Web mode. This parameter is ignored if an SSL connection is not made.
- Use the `Client certificate` key field to specify the file containing the secret key used for the client certificate. This file will replace the default `~/.postgresql/postgresql.key` if PEM is installed in Desktop mode, and `<STORAGE_DIR>/<USERNAME>/.postgresql/postgresql.key` if PEM is installed in Web mode. This parameter is ignored if an SSL connection is not made.
- Use the `Root certificate` field to specify the file containing the SSL certificate authority. This file will replace the default `~/.postgresql/root.crt` file. This parameter is ignored if an SSL connection is not made.
- Use the `Certificate revocation list` field to specify the file containing the SSL certificate revocation list. This list will replace the default list, found in `~/.postgresql/root.crl` . This parameter is ignored if an SSL connection is not made.
- When `SSL compression?` is set to True, data sent over SSL connections will be compressed. The default value is `False` (compression is disabled). This parameter is ignored if an SSL connection is not made.

Warning

Certificates, private keys, and the revocation list are stored in the per-user file storage area on the server, which is owned by the user account under which the PEM server process is run. This means that administrators of the server may be able to access those files; appropriate caution should be taken before choosing to use this feature.

The SSH Tunnel tab of the Create – Server dialog

Use the fields on the `SSH Tunnel` tab to configure SSH Tunneling. You can use a tunnel to connect a database server (through an intermediary proxy host) to a server that resides on a network to which the client may not be able to connect directly.

- Set `Use SSH tunneling` to `Yes` to specify that PEM should use an SSH tunnel when connecting to the specified server.
- Specify the name or IP address of the SSH host (through which client connections will be forwarded) in the `Tunnel host` field.
- Specify the port of the SSH host (through which client connections will be forwarded) in the `Tunnel port` field.
- Specify the name of a user with login privileges for the SSH host in the `Username` field.
- Specify the type of authentication that will be used when connecting to the SSH host in the `Authentication` field.
- Select `Password` to specify that PEM will use a password for authentication to the SSH host. This is the default.

- Select `Identity file` to specify that PEM will use a private key file when connecting.
- If the SSH host is expecting a private key file for authentication, use the `Identity file` field to specify the location of the key file.
- If the SSH host is expecting a password, use the `Password` field to specify the password, or if an identity file is being used, the passphrase.

The Advanced tab of the Create – Server dialog

Use fields on the `Advanced` tab to specify details that are used to manage the server:

- Specify the IP address of the server host in the `Host Address1` field.
- Use the `DB restriction` field to specify a SQL restriction that will be used against the pg_database table to limit the databases displayed in the tree control. For example, you might enter: `'live_db'` , `'test_db'` to instruct the PEM browser to display only the `live_db` and `test_db` databases. Note that you can also limit the schemas shown in the database from the database properties dialog by entering a restriction against pg_namespace.
- Use the `Password file` field to specify the location of a password file (`.pgpass`). The `.pgpass` file allows a user to login without providing a password when they connect. For more information, see the Postgres documentation at:

  http://www.postgresql.org/docs/current/static/libpq-pgpass.html
- Use the `Service ID` field to specify parameters to control the database service process. For servers that are stored in the Enterprise Manager directory, enter the service ID. On Windows machines, this is the identifier for the Windows service. On Linux machines, the name of the init script used to start the server is `/etc/init.d` and the name of the systemd script to start the server is systemctl. For example, the name of the Advanced Server 10 service is `edb-as-10` . For local servers, the setting is operating system dependent:
  - If the PEM client is running on a Windows machine, it can control the postmaster service if you have sufficient access rights. Enter the name of the service. In case of a remote server, it must be prepended by the machine name (e.g. `PSE1\pgsql-8.0`). PEM will automatically discover services running on your local machine.
  - If the PEM client is running on a Linux machine, it can control processes running on the local machine if you have enough access rights. Provide a full path and needed options to access the `pg_ctl` program. When executing service control functions, PEM will append status/start/stop keywords to this. For example:

    `sudo /usr/pgsql-x/bin/pg_ctl -D /var/lib/pgsql/x/data` where `x` is the version of the PostgreSQL database server.
- If the server is a member of a Failover Manager cluster, you can use PEM to monitor the health of the cluster and to replace the master node if necessary. To enable PEM to monitor Failover Manager, use the `EFM cluster name` field to specify the cluster name. The cluster name is the prefix of the name of the Failover Manager cluster properties file. For example, if the cluster properties file is named `efm.properties` , the cluster name is `efm` .
- If you are using PEM to monitor the status of a Failover Manager cluster, use the `EFM installation path` field to specify the location of the Failover Manager binary file. By default, the Failover Manager binary file is installed in `/usr/edb/efm-x.x/bin` , where `x.x` specifies the Failover Manager version.

The PEM Agent tab of the Create – Server dialog

Use fields on the `PEM Agent` tab to specify connection details for the PEM agent:

- Select an Enterprise Manager agent using the drop-down listbox to the right of the `Bound agent` label. One agent can monitor multiple Postgres servers.
- Move the `Remote monitoring?` slider to `Yes` to indicate that the PEM agent does not reside on the same host as the monitored server. When remote monitoring is enabled, agent level statistics for the monitored server will not be available for custom charts and dashboards, and the remote server will not

be accessible by some PEM utilities (such as Audit Manager, Capacity Manager, Log Manager, Postgres Expert and Tuning Wizard).

- Enter the IP address or socket path that the agent should use when connecting to the database server in the `Host` field. By default, the agent will use the host address shown on the `General` tab. On a Unix server, you may wish to specify a socket path, e.g. `/tmp` .
- Enter the `Port` number that the agent will use when connecting to the server. By default, the agent will use the port defined on the `Properties` tab.
- Use the drop-down listbox in the `SSL` field to specify an SSL operational mode; specify require, prefer, allow, disable, verify-ca or verify-full. For more information about using SSL encryption, see the PostgreSQL documentation at:

  https://www.enterprisedb.com/edb-docs/d/postgresql/reference/manual/12.1/libpq-ssl.html
- Use the `Database` field to specify the name of the database to which the agent will initially connect.
- Specify the name of the role that agent should use when connecting to the server in the `User name` field. Note that if the specified role is not a database superuser, then some of the features will not work as expected. For the list of features that do not work if the specified role is not a database superuser, see Agent privileges.

  If you are using Postgres version 10 or above, you can use the `pg_monitor` role to grant the required privileges to a non-superuser. For information about `pg_monitor` role, see:

  https://www.postgresql.org/docs/current/default-roles.html
- Specify the password that the agent should use when connecting to the server in the `Password` field, and verify it by typing it again in the `Confirm password` field. If you do not specify a password, you will need to configure the authentication for the agent manually; for example, you can use a `.pgpass` file.
- Set the `Allow takeover?` slider to `Yes` to specify that the server may be taken over by another agent. This feature allows an agent to take responsibility for the monitoring of the database server if, for example, the server has been moved to another host as part of a high availability failover process.

The Create Server dialog (BART - General tab)

Use the fields on the `General` tab under `BART` tab to describe the general properties of the BART Server that will map to the PEM server:

- Use the `BART server` field to select the BART server name. All the BART servers configured in the PEM console will be listed in this drop down list.
- Use the `Server name` field to specify a name for the database server that you want to backup using the BART server. This name gets stored in the BART configuration file.
- Use the `Backup name` field to specify a template for user-defined names to be assigned to the backups of the database server. If you do not specify a backup name template, then the backup can only be referenced in BART sub-commands by the BART assigned, integer backup identifier.
- Use the `Host address` field to specify the IP address of the database server that you want to configure for backup.
- Use the `Port` field to specify the port to be used for the database that you want to backup.
- Use the `User` field to specify the user of the database that you want to backup using BART through PEM console. If you want to enable incremental backups for this database server, then the user must be a superuser.
- Use the `Password` field to specify the password for the user of the database that you want to backup.
- Use the `Cluster Owner` field to specify the Linux operating system user account that owns the database cluster. This is typically `enterprisedb` for Advanced Server database clusters installed in the Oracle databases compatible mode, or `postgres` for PostgreSQL database clusters and for Advanced Server database clusters installed in the PostgreSQL databases compatible mode.
- Use the `Archive command` field to specify the desired format of the archive command string to be used in the `bart.cfg` file. Inputs provided for the Archive command will overwrite the database server's

`Postgresql.conf` file. Once the server gets added, the database server will be restarted or database configurations will be reloaded.
- Use the `Allow incremental backup?` switch to specify if incremental backup should be enabled for this database server.
- Use the `Setup passwordless SSH?` switch to specify if you want to create SSH certificates to allow passwordess logins between the Database Server and the BART server. Ensure to bind a PEM agent before setting up the passwordless SSH authentication. Passwordless SSH will not work for a database server being remotely monitored by a PEM agent.

The Create - Server dialog (BART - Misc tab)

Use the fields on the `Misc` tab under `BART` tab to describe the miscellaneous properties of the BART Server:

- Use the `Override default configuration?` Switch to specify if you want to override the BART server configurations with the specific database server configurations.
- Use the `Xlog` method to specify how the transaction log should be collected during the execution of `pg_basebackup`.
- Use the `Retention policy` field to specify the retention policy for the backup. This determines when an active backup should be marked as obsolete, and hence, be a candidate for deletion. You can specify the retention policy in terms of number of backup or in terms of duration (days, weeks, or months).
- Use the `WAL compression` switch to specify if you want to compress the archived Xlog/WAL files in Gzip format. To enable WAL compression, the gzip compression program must be present in the BART user account's PATH. The wal_compression setting must not be enabled for those database servers where you need to take incremental backups.
- Use the `Copy WALs during restore` field to specify how the archived WAL files are collected when invoking the RESTORE operation. Set to enabled to copy the archived WAL files from the BART backup catalog to the <restore_path>/archived_wals directory prior to the database server archive recovery. Set to disabled to retrieve the archived WAL files directly from the BART backup catalog during the database server archive recovery.
- Use the `Thread count` field to specify the number of threads to copy the blocks. You must set `thread count` to `1` if you want to take a backup with the `pg_basebackup` utility.
- Use the `Batch size` field to specify the number of blocks of memory used for copying modified blocks, applicable only for incremental backups.
- Use the `Scan interval` field to specify the number of seconds after which the WAL scanner should scan the new WAL files.
- Use the `MBM scan timeout` field to specify the number of seconds to wait for MBM files before timing out, applicable only for incremental backups.

To view the properties of a server, right-click on the server name in the PEM client tree control, and select the `Properties…` option from the context menu. To modify a server's properties, disconnect from the server before opening the `Properties` dialog.

Automatic Server Discovery

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

To enable auto discovery for a specific agent, you must enable the `Server Auto Discovery` probe. To access the Manage Probes tab, highlight the name of a PEM agent in the PEM client tree control, and select `Manage Probes...` from the `Management` menu. When the `Manage Probes` tab opens, confirm that the slider control in the `Enabled?` column is set to `Yes`.

To open the `Auto Discovery` dialog, highlight the name of a PEM agent in the PEM client tree control, and select `Auto Discovery...` from the `Management` menu.

The PEM Auto Discovery dialog

When the `Auto Discovery` dialog opens, the `Discovered Database Servers` box will display a list of servers that are currently not being monitored by a PEM agent. Check the box next to a server name to display information about the server in the `Server Connection Details` box, and connection properties for the agent in the `Agent Connection Details` box.

Use the `Check All` button to select the box next to all of the displayed servers, or `Uncheck All` to deselect all of the boxes to the left of the server names.

The fields in the `Server Connection Details` box provide information about the server that PEM will monitor:

- Accept or modify the name of the monitored server in the `Name` field. The specified name will be displayed in the tree control of the PEM client.
- Use the `Server group` drop-down listbox to select the server group under which the server will be displayed in the PEM client tree control.
- Use the `Host name/address` field to specify the IP address of the monitored server.
- The `Port` field displays the port that is monitored by the server; this field may not be modified.
- Provide the name of the service in the `Service ID` field. Please note that the service name must be provided to enable some PEM functionality.
- By default, the `Maintenance database` field indicates that the selected server uses a Postgres maintenance database. Customize the content of the `Maintenance database` field for your installation.

The fields in the `Agent Connection Details` box specify the properties that the PEM agent will use when connecting to the server:

- The `Host` field displays the IP address that will be used for the PEM agent binding.
- The `User name` field displays the name that will be used by the PEM agent when connecting to the selected server.
- The `Password` field displays the password associated with the specified user name.
- Use the drop-down listbox in the `SSL mode` field to specify your SSL connection preferences.

When you've finished specifying the connection properties for the servers that you are binding for monitoring, click the `OK` button to register the servers. Click `Cancel` to exit without preserving any changes.

The registered server

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

Using the pemworker Utility to Register a Server

You can use the `pemworker` utility to register a server for monitoring by the PEM server or to unregister a database server. During registration, the `pemworker` utility will bind the new server to the agent that resides on the system from which you invoked the registration command. To register a server:

on a Linux host, use the command:

```
pemworker --register-server
```

on a Windows host, use the command:

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

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

Option	Description
`--pem-user`	Specifies the name of the PEM administrative user. Required.
`--server-addr`	Specifies the IP address of the server host, or the fully qualified domain name. On Unix ba
`--server-port`	Specifies the port number of the host. Required.
`--server-database`	Specifies the name of the database to which the server will connect. Required.

Option	Description
`--server-user`	Specify the name of the user that will be used by the agent when monitoring the server. Re
`--server-service-name`	Specifies the name of the database service that controls operations on the server that is be
`--remote-monitoring`	Include the --remote-monitoring clause and a value of no (the default) to indicate that the s
`--efm-cluster-name`	Specifies the name of the Failover Manager cluster that monitors the server (if applicable).
`--efm-install-path`	Specifies the complete path to the installation directory of Failover Manager (if applicable).
`--asb-host-name`	Specifies the name of the host to which the agent is connecting.
`--asb-host-port`	Specifies the port number that the agent will use when connecting to the database.
`--asb-host-db`	Specifies the name of the database to which the agent will connect.
`--asb-host-user`	Specifies the database user name that the agent will supply when authenticating with the c
`--asb-ssl-mode`	Specifies the type of SSL authentication that will be used for connections. Supported value
`--group`	Specifies the name of the group in which the server will be displayed.
`--team`	Specifies the name of the group role that will be allowed to access the server.
`--owner`	Specifies the name of the role that will own the monitored server.

Set the environment variable `PEM_SERVER_PASSWORD` to provide the password for the PEM server to allow the pemworker to connect as a PEM admin user.

Set the environment variable `PEM_MONITORED_SERVER_PASSWORD` to provide the password of the database server being registered and monitored by pemagent.

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

Using the pemworker Utility to Unregister a Server

You can use the `pemworker` utility to unregister a database server; to unregister a server, invoke the `pemworker` utility:

on a Linux host, use the command:

```
pemworker --unregister-server
```

on a Windows host, use the command:

```
pemworker.exe UNREGISTER-SERVICE
```

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

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

Set environment variable PEM_SERVER_PASSWORD to provide the password for the PEM server to allow the pemworker to connect as a PEM admin user.

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

Verifying the Connection and Binding

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

The Global Overview dashboard

When initially connecting to a newly bound server, the `Global Overview` dashboard may display the new server with a status of "unknown" in the server list; before recognizing the server, the bound agent must execute a number of probes to examine the server, which may take a few minutes to complete depending on network availability.

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

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

1.3 Managing Certificates

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

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

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

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

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

The following sections contain detailed information about manually replacing certificate files.

replacing ssl certificates ssl certificates, replacing

Replacing SSL Certificates

The following steps detail replacing the SSL certificates on an existing PEM installation. If you plan to upgrade your server to a new version at the same time, invoke all of the PEM installers (first the server installer, then agent installers) before replacing the SSL certificates. Then:

1. Stop all running PEM agents, first on the server host, and then on any monitored node.

   To stop a PEM agent on a Linux host, open a terminal window, assume superuser privileges, and enter the command:

   On Linux with init.d, for eg: Centos6

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

   On Linux with systemd, for eg: Centos7

   ```
   systemctl stop pemagent
   ```

   On a Windows host, you can use the `Services` applet to stop the PEM agent. The PEM agent service is named Postgres Enterprise Manager Agent; highlight the service name in the `Services` dialog, and click `Stop the service`.

2. Take a backup of the existing SSL keys and certificates. The SSL keys and certificates are stored in the `data` directory under your PEM installation. For example, the default location on a Linux system is:

   `/var/lib/pgsql/x/data` where `x` is the PostgreSQL database version.

   Make a copy of the following files, adding an extension to each file to make the name unique:

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

   For example, the command:

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

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

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

   ```
   /usr/pgsql-x.x/bin/psql -U postgres -d pem --no-psqlrc -t -A -c "SELECT public.openssl_rs
   ```

   After creating the `ca_key.key` file, `cat` the contents to the variable `CA_KEY` for use when generating the `ca_certificate.crt` file and modify the privileges on the `ca_key.key` file:

   ```
   CA_KEY=$(cat /var/lib/pgsql/x/data/ca_key.key)
   ```

   ```
   chmod 600 /var/lib/pgsql/x/data/ca_key.key
   ```

4. Use the key to generate the `ca_certificate.crt` file. For simplicity, place the SQL query into a temporary file with a unique name:

   ```
   echo "SELECT openssl_csr_to_crt(openssl_rsa_key_to_csr('${CA_KEY}', 'PEM','US', 'MA', 'Be
   ```

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

   ```
   /usr/pgsql-x/bin/psql -U postgres -d pem --no-psqlrc -t -A -f /tmp/_random.$$ > /var/lib/
   ```

   Modify the permissions of the `ca_certificate.crt` file, and remove the temporary file that contained the SQL command:

   ```
   chmod 600 /var/lib/pgsql/x/data/ca_certificate.crt
   ```

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

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

   ```
   cp /var/lib/pgsql/x/data/ca_certificate.crt /var/lib/pgsql/x/data/root.crt
   ```

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

   ```
   chmod 600 /var/lib/pgsql/x/data/root.crt
   ```

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

   ```
   /usr/pgsql-x/bin/psql -U postgres -d pem --no-psqlrc -t -A -c "SELECT openssl_rsa_generat
   ```

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

```
chmod 600 /var/lib/pgsql/x/data/root.crl
```

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

   ```
   /usr/pgsql-x/bin/psql -U postgres -d pem --no-psqlrc -t -A -c "SELECT public.openssl_rsa_
   ```

   After creating the `server.key` file, `cat` the contents to the variable `SSL_KEY` for use when generating the `server.crt` file and modify the privileges on the `server.key` file:

   ```
   SSL_KEY=$(cat /var/lib/pgsql/x/data/server.key)
   ```

   ```
   chmod 600 /var/lib/pgsql/x/data/server.key
   ```

8. Use the `SSL_KEY` to generate the server certificate. Save the certificate in the `server.crt` file. For simplicity, first place the SQL query into a temporary file with a unique name:

   ```
   echo "SELECT openssl_csr_to_crt(openssl_rsa_key_to_csr('${SSL_KEY}', 'PEM','US', 'MA', 'B
   ```

   ```
   /usr/pgsql-x/bin/psql -U postgres -d pem --no-psqlrc -t -A -f /tmp/_random.$$ >> /var/lib
   ```

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

   ```
   chmod 600 /var/lib/pgsql/x/data/server.crt
   ```

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

10. Restart the Postgres server:

    On Linux with init.d, for eg: Centos6

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

    On Linux with systemd, for eg: Centos7

    ```
    systemctl restart postgresql-x
    ```

agent ssl certificate update agent ssl certificate

Updating Agent SSL Certificates

For each agent that interacts with the PEM server, you must:

- generate an rsa key and a certificate.
- copy the key and certificate to the agent.
- restart the agent.

Each agent has a unique identifier that is stored in the `pem.agent` table in the `pem` database. You must replace the key and certificate files with the key or certificate that corresponds to the agent's identifier. Please note that you must move the `agent.key` and `agent.crt` files (generated in Steps 2 and 3 into place on their respective PEM agent host before generating the next key file pair; subsequent commands will overwrite the previously generated file.

To generate a PEM agent key file pair:

1. Use psql to find the number of agents and their corresponding identifiers:

   ```
   /usr/pgsql-x/bin/psql -U postgres -d pem --no-psqlrc -t -A -c "SELECT ID FROM pem.agent"
   ```

- On Linux, you can also find the agent identifier and location of the keys and certificates in the `PEMagent` section of the `/etc/postgres-reg.ini` file.
- On Windows, the information is stored in the registry:
  - On a 64-bit Windows installation, check:

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

  - On a 32-bit Windows installation, check:

    `HKEY_LOCAL_MACHINE\SOFTWARE\EnterpriseDB\PEM\agent`

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

   ```
   /usr/pgsql-x/bin/psql -U postgres -d pem --no-psqlrc -t -A -c "SELECT openssl_rsa_generat
   ```

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

   ```
   chmod 600 agent.key
   ```

3. Generate a certificate for each agent. To generate a certificate, execute the following command, capturing the output in a certificate file:

   ```
   /usr/pgsql-x/bin/psql -U postgres -d pem --no-psqlrc -t -A -c "SELECT openssl_csr_to_crt(
   ```

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

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

   ```
   chmod 600 agent.crt
   ```

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

- On Linux with init.d, restart the service with the command:

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

  On Linux with systemd, restart the service with the command:

  ```
  systemctl start pemagent
  ```

- On a Windows host, you can use the Services applet to start the PEM agent. The PEM agent service is named `Postgres Enterprise Manager` Agent; highlight the service name in the Services dialog, and click `Start the service` .

1.4 Managing Configuration Settings

Multiple configuration files are read at startup by Postgres Enterprise Manager. The files are as follows:

- `config.py` : This is the main configuration file, and should not be modified. It can be used as a reference for configuration settings, that may be overridden in one of the following files.
- `config_distro.py` : This file is read after `config.py` and is intended for packagers to change any settings that are required for their Postgres Enterprise Manager distribution. This may typically include certain paths and file locations. This file is optional, and may be created by packagers in the same directory as `config.py` if needed.
- `config_local.py` : This file is read after `config_distro.py` and is intended for end users to change any default or packaging specific settings that they may wish to adjust to meet local preferences or standards.This file is optional, and may be created by users in the same directory as `config.py` if needed.

A copy of the default `config.py` file is included in the PEM online help for reference.

1.5 Managing a PEM Server

The sections that follow provide information about tasks related to PEM server such as restarting the PEM server and agent, controlling the PEM server or PEM agent, controlling the HTTPD service on Linux and Windows, controlling the HTTPD server, managing PEM authentication and security, modifying the `pg_hba.conf` file, modifying PEM to use a proxy server etc.

starting service restart PEM service restart PEM agent

Starting and Stopping the PEM Server and Agents

The PEM server starts, stops and restarts when the Postgres server instance on which it resides starts, stops or restarts; use the same commands to control the PEM server that you would use to control the Postgres server. On Linux platforms, the command that stops and starts the service script will vary by platform and OS version.

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

The Windows operating system includes a graphical service controller that displays the server status, and offers point-and-click server control. The `Services` utility can be accessed through the Windows `Control Panel` . When the utility opens, use the scroll bar to navigate through the listed services to highlight the service name.

The PEM service in the Windows Services window

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

Please note that any user (or client application) connected to the Postgres server will be abruptly disconnected if you stop the service. For more information about controlling a service, please consult the *EDB Postgres Advanced Server Installation Guide*, available from the EnterpriseDB website at:

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

control server remotely

Remotely Starting and Stopping Monitored Servers

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

- specify the `Store on PEM Server` option on the `Properties` dialog.
- specify the name of a service script in the `Service ID` field on the `Advanced` tab:
  - For Advanced Server, the service name is `edb-as-<x>` or `ppas-<x>` .
  - For PostgreSQL, the service name is `postgresql-<x>` .

  Where *x* indicates the server version number.

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

The Management menu of a managed server

Control service on Linux

Controlling the PEM Server or PEM Agent on Linux

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

- a PEM server on Advanced Server is `edb-as-<x>` or `ppas-<x>`
- a PEM server on PostgreSQL is `postgresql-<x>`
- a PEM agent is `pemagent`

Where *x* indicates the server version number.

You can use the service script to control the service.

- To control a service on RHEL or CentOS version 6.x, open a command line, assume superuser privileges, and enter:

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

- To control a service on RHEL or CentOS version 7.x, open a command line, assume superuser privileges, and issue the command:

  ```
  systemctl <service_name> <action>
  ```

Where:

service_name is the name of the service.

action specifies the action taken by the service. Specify:

- `start` to start the service.
- `stop` to stop the service.
- `restart` to stop and then start the service.
- `status` to check the status of the service.

Control service on Windows

Controlling the PEM Server or PEM Agent on Windows

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

- a PEM server host on PostgreSQL is `postgresql-<x>`
- a PEM server host on Advanced Server is `edb-as-<x>` , or `ppas-<x>`
- a PEM agent is `Postgres Enterprise Manager - pemAgent`

Where *x* indicates the server version number.

Navigate through the Windows `Control Panel` to open the `Services` utility. When the utility opens, use the scroll bar to browse the list of services.

The Windows Services window

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

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

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

HTTPD server

Controlling the HTTPD Server

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

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

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

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

To start the service on a CentOS or RHEL 6.x system, use the command:

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

To start the service on a CentOS or RHEL 7.x system, use the command:

```
systemctl start httpd
```

On Windows, you can use the `Services` applet to check the status of the `PEM HTTPD` service. After opening the Services applet, scroll through the list to locate the `PEM HTTPD` service.

The PEM HTTPD Windows service

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

pg_hba.conf file

Modifying the pg_hba.conf File

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

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

PEM entries in the pg_hba.conf file

You may also need to manually modify the `pg_hba.conf` file to allow connections between the PEM server and other components. For example, if your PEM-HTTPD installation does not reside on the same host as the PEM server, you must modify the `pg_hba.conf` file on the PEM server host to allow PEM-HTTPD to connect to the server.

By default, the `pg_hba.conf` file resides in the data directory, under your Postgres installation; for example, on an Advanced Server 10 host, the default location of the pg_hba.conf is:

```
    /var/lib/edb/as10/data/pg_hba.conf
```

You can modify the `pg_hba.conf` file with your editor of choice. After modifying the file, restart the server for changes to take effect.

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

```
---
title: "TYPE     DATABASE        USER        CIDR-ADDRESS          METHOD"
1.5 TYPE     DATABASE        USER        CIDR-ADDRESS          METHOD
---

<div id="managing_pem_server" class="registered_link"></div>

---
title: "IPv4 local connections:"
1.5 IPv4 local connections:
---

<div id="managing_pem_server" class="registered_link"></div>

 host     postgres       postgres    192.168.10.102/32       md5
```

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

```
---
title: "TYPE     DATABASE        USER        CIDR-ADDRESS        METHOD"
1.5 TYPE     DATABASE        USER        CIDR-ADDRESS       METHOD
---
```

```
<div id="managing_pem_server" class="registered_link"></div>

---
title: "IPv4 local connections:"
1.5 IPv4 local connections:
---

<div id="managing_pem_server" class="registered_link"></div>
```

```
  host      postgres         all       192.168.10.0/24       md5
```

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

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

Before you can connect to a Postgres server with PEM, you must ensure that the `pg_hba.conf` file on both servers allows the connection.

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

Creating and Maintaining Databases and Objects

Each instance of a Postgres server manages one or more databases; each user must provide authentication information to connect to the database before accessing the information contained within it. The PEM client provides dialogs that allow you to create and manage databases, and all of the various objects that comprise a database (e.g. tables, indexes, stored procedures, etc.).

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

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

Use PEM's dialogs to create and manage database objects

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

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

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

Managing PEM Authentication

Postgres supports a number of authentication methods:

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

- PAM

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

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

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

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

```
localhost:5432:*:postgres:1safepwd
```

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

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

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

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

Editing the PEM Server Configuration

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

The Server Configuration dialog

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

Security

Managing Security

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

Login Roles

Login Roles

When you connect to the PEM server, you must provide role credentials that allow access to the database on which the PEM server stores data. By default, the postgres superuser account is used to initially connect to the server, but it is strongly recommended (for both security and auditing purposes) that individual roles are created for each connecting user. You can use the PEM Query Tool, the PEM web interface `Create - Login/Group Role` dialog, or a command line client (such as psql) to create a role.

To use the `Create - Login/Group Role` dialog to create a role, expand the node for the server on which the role will reside in the PEM tree control, and right-click on the `Login/Group Roles` node to access the context menu. Then, select `Login/Group Role...` from the `Create` menu.

The context menu of the Login Roles node

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

When you've finished defining the new role, click `Save` to create the role.

The Login Role dialog

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

For more complete information about creating and managing a role, see the PostgreSQL online documentation:

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

Group Roles

Group Roles

Group roles can serve as containers, used to dispense system privileges (such as creating databases) and object privileges (e.g. inserting data into a particular table). The primary purpose of a group role is to make the mass management of system and object permissions much easier for a DBA. Rather than assigning or modifying privileges individually across many different login accounts, you can assign or change privileges for a single role and then grant that role to many login roles at once.

Use the `Group Roles` node (located beneath the name of each registered server in the PEM tree control) to create and manage group roles. Options on the context menu provide access to a dialog that allows you to create a new role or modify the properties of an existing role. You can find more information about creating roles at:

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

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

You can use the `Login/Group Role` dialog to allow a role with limited privileges to access PEM features such as the Audit Manager, Capacity Manager, or SQL Profiler. PEM pre-defined roles allow access to PEM functionality; roles that are assigned membership in these roles can access the associated feature.

The Membership tab

When defining a user, use the `Membership` tab to specify the roles in which the new user is a member. The new user will share the privileges associated with each role in which it is a member. For a user to have access to PEM extended functionality, the role must be a member of the pem_user role and the pre-defined role that grants access to the feature. Use the `Roles` field to select pre-defined role names from a drop down list.

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

The SQL tab

The example shown above creates a login role named `acctg_clerk` that will have access to the `Audit Manager` ; the role can make unlimited connections to the server at any given time.

You can use PEM pre-defined roles to allow access to the functionality listed in the table below:

Value	Parent Role	Description
pem_super_admin		Role to manage/configure everything on Postgre
pem_admin	pem_super_admin	Role for administration/management/configuratio
pem_config	pem_admin	Role for configuration management of Postgres
pem_component	pem_admin	Role to run/execute all wizard/dialog based com
pem_rest_api	pem_admin	Role to access the REST API.
pem_server_service_manager	pem_admin	Role for allowing to restart/reload the monitored
pem_manage_schedule_task	pem_admin	Role to configure the schedule tasks.
pem_manage_alert	pem_admin	Role for managing/configuring alerts, and its ten
pem_config_alert	pem_config, pem_manage_alert	Role for configuring the alerts on any monitored

Value	Parent Role	Description
pem_manage_probe	pem_admin	Role to create, update, delete the custom probes
pem_config_probe	pem_config, pem_manage_probe	Role for probe configuration (history retention, ex
pem_database_server_registration	pem_admin	Role to register a database server.
pem_comp_postgres_expert	pem_component	Role to run the Postgres Expert.
pem_comp_auto_discovery	pem_component	Role to run the Auto discovery of a database ser
pem_comp_log_analysis_expert	pem_component	Role to run the Log Analysis Expert.
pem_comp_sqlprofiler	pem_component	Role to run the SQL Profiler.
pem_manage_efm	pem_admin	Role to manage Failover Manager functionality.
pem_comp_capacity_manager	pem_component	Role to run the Capacity Manager.
pem_comp_log_manager	pem_component	Role to run the Log Manager.
pem_comp_audit_manager	pem_component	Role to run the Audit Manager.
pem_comp_tuning_wizard	pem_component	Role to run the Tuning Wizard.

Using a Team Role

When you register a server for monitoring by PEM, you can specify a *Team* that will be associated with the server. A Team is a group role that can be used to allow or restrict access to one or more monitored servers to a limited group of role members. The PEM client will only display a server with a specified Team to those users who are:

- a member of the Team role
- the role that created the server
- a role with superuser privileges on the PEM server.

To create a team role, expand the node for the server on which the role will reside in the PEM tree control, and right-click on the `Login/Group Roles` node to access the context menu. Then, select `Login/Group Role…` from the `Create` menu; when the `Create - Login/Group Role` dialog opens, use the fields provided to specify the properties of the team role.

Object Permissions

A role must be granted sufficient privileges before accessing, executing, or creating any database object. PEM allows you to assign (`GRANT`) and remove (`REVOKE`) object permissions to group roles or login accounts using the graphical interface of the PEM client.

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

The PEM client also contains a `Grant Wizard` (accessed through the `Tools` menu) that allows you to manage many object permissions at once.

Managing Job Notifications

Sending email notifications for a job

You can configure the settings in PEM console for sending the SMTP trap on success or failure of a system-generated job (listed under scheduled tasks) or a custom-defined agent job. For information about custom-defined agent jobs, see ‘Creating PEM Scheduled Jobs’. These email notification settings can be configured at following three levels (in order of precedence) to send email notifications to the specified user group:

- Job level
- Agent level
- PEM server level (default level)

Configuring Job Notifications at Job Level

You can configure email notification settings at job level only for a custom-defined agent job in one of the following ways:

- For a new agent job, you can configure the email notification settings in the *Notification* tab of *Create-Agent Job* wizard while creating the job itself.
- For an existing custom-defined job, you can edit the properties of the job and configure the notification settings.

Job notification configuration: job level

Use the fields on the *Notifications* tab to configure the email notification settings on job level:

- Use the *Send the notifications* field to specify when you want the email notifications to be sent.
- Use the *Email group* field to specify the email group that should receive the email notification.

Configuring Job Notifications at Agent Level

Select the agent in the tree view, right click and select *Properties*. In the Properties dialog, select the *Job notifications* tab.

Job notification configuration: agent level

Use the fields on the Job notifications tab to configure the email notification settings on agent level:

- Use the *Override default configuration?* switch to specify if you want the agent level job notification settings to override the default job notification settings. If you select *Yes* for this switch, you can use the rest of the settings on this dialog to define when and to whom the job notifications should be sent. Please note that the rest of the settings on this dialog work only if you enable the *Override default configuration?* switch.
- Use the *Email on job completion?* switch to specify if the job notification should be sent on the successful job completion.
- Use the *Email on a job failure?* switch to specify if the job notification should be sent on the failure of a job.
- Use the *Email group* field to specify the email group to whom the job notification should be sent.

Configuring Job Notifications at Server Level

You can use the *Server Configuration* dialog to provide information about your email notification configuration at PEM server level. To open the Server Configuration dialog, select *Server Configuration...* from the PEM client's Management menu.

Job notification configuration: server level

Four server configuration parameters specify information about your job notification preferences at PEM server level:

- Use the *job_failure_notification* switch to specify if you want to send email notification after each job failure.
- Use the *job_notification_email_group* parameter to specify the email group that should receive the email notification.
- Use the *job_retention_time parameter* to specify the number of days that non-recurring scheduled tasks should be retained in the system.
- Use the *job_status_change_notification* switch to specify if you want to send email notification after each job status change, irrespective of its status being a failure, success, or interrupted.

Managing PEM Scheduled Jobs

Creating PEM scheduled jobs

You can create a PEM scheduled job to perform a set of custom-defined steps in the specified sequence. These steps may contain SQL code or a batch/shell script that you may run on a server that is bound with the agent. You can schedule these jobs to suit your business requirements. For example, you can create a job for taking a backup of a particular database server and schedule it to run on a specific date and time of every month.

To create or manage a PEM scheduled job, use the PEM tree control to browse to the PEM agent for which you want to create the job. The tree control will display a Jobs node, under which currently defined jobs are displayed. To add a new job, right click on the `Jobs` node, and select `Create Job...` from the context menu.

When the `Create - Agent Job` dialog opens, use the tabs on the `Create - Agent Job` dialog to define the steps and schedule that make up a PEM scheduled job.

PEM scheduled job dialog create schedule

Use the fields on the `General` tab to provide general information about a job:

- Provide a name for the job in the `Name` field.
- Move the `Enabled` switch to the `Yes` position to enable a job, or `No` to disable a job.
- Use the `Comment` field to store notes about the job.

PEM scheduled job dialog add steps

Use the `Steps` tab to define and manage the steps that the job will perform. Click the Add icon (+) to add a new step; then click the compose icon (located at the left side of the header) to open the step definition dialog:

PEM scheduled job steps definition

Use fields on the step definition dialog to define the step:

- Provide a name for the step in the `Name` field; please note that steps will be performed in alphanumeric order by name.
- Use the `Enabled` switch to include the step when executing the job (`True`) or to disable the step (`False`).
- Use the `Kind` switch to indicate if the job step invokes SQL code (`SQL`) or a batch script (`Batch`).
  - If you select `SQL` , use the `Code` tab to provide SQL code for the step.
  - If you select `Batch` , use the `Code` tab to provide the batch script that will be executed during the step.
- Use the `On error` drop-down to specify the behavior of pgAgent if it encounters an error while executing the step. Select from:
  - Fail - Stop the job if you encounter an error while processing this step.
  - Success - Mark the step as completing successfully, and continue.
  - Ignore - Ignore the error, and continue.
- If you have selected SQL as your input for `Kind` switch, provide the following additional information:
  - Use the `Server` field to specify the server that is bound with the agent for which you are creating the PEM scheduled job.
  - Use the `Database` field to specify the database that is associated with the server that you have selected.
- Use the `Comment` field to provide a comment about the step.

PEM scheduled job steps definition code

- Use the context-sensitive field on the step definition dialog's `Code` tab to provide the SQL code or batch script that will be executed during the step:
  - If the step invokes SQL code, provide one or more SQL statements in the `SQL query` field.
  - If the step invokes a batch script, provide the script in the `Code` field. If you are running on a Windows server, standard batch file syntax must be used. When running on a Linux server, any shell script may be used, provided that a suitable interpreter is specified on the first line (e.g. *#!/bin/sh*). Along with the defined inline code, you can also provide the path of any batch script, shell script, or SQL file on the filesystem.
    To invoke a script on a Linux system, you must modify the entry for *batch_script_user* parameter of agent.cfg file and specify the user that should be used to run the script. You can either specify a non-root user or root for this parameter. If you do not specify a user, or the specified user does not exist, then the script will not be executed. Restart the agent after modifying the file.
    To invoke a script on a Windows system, set the registry entry for *AllowBatchJobSteps* as true and restart the PEM agent. PEM registry entries are located in `HKEY_LOCAL_MACHINE\Software\Wow6432Node\E`
    .

After providing all the information required by the step, click the `Save` button to save and close the step definition dialog.

Click the add icon (+) to add each additional step, or select the `Schedules` tab to define the job schedule.

Click the Add icon (+) to add a schedule for the job; then click the compose icon (located at the left side of the header) to open the schedule definition dialog:

PEM scheduled job dialog add schedule tab

Use the fields on the `Schedules definition` tab to specify the days and times at which the job will execute.

- Provide a name for the schedule in the `Name` field.
- Use the *Enabled* switch to indicate that pgAgent should use the schedule (`Yes`) or to disable the schedule (`No`).
- Use the calendar selector in the `Start` field to specify the starting date and time for the schedule.
- Use the calendar selector in the `End` field to specify the ending date and time for the schedule.
- Use the `Comment` field to provide a comment about the schedule.

Select the `Repeat` tab to define the days on which the schedule will execute.

PEM scheduled job dialog schedule repeat tab

Use the fields on the `Repeat` tab to specify the details about the schedule in a cron-style format. The job will execute on each date or time element selected on the `Repeat` tab.

Click within a field to open a list of valid values for that field; click on a specific value to add that value to the list of selected values for the field. To clear the values from a field, click the X located at the right-side of the field.

- Use the fields within the `Days` box to specify the days on which the job will execute:
  - Use the `Week Days` field to select the days on which the job will execute.
  - Use the `Month Days` field to select the numeric days on which the job will execute. Specify the `Last Day` to indicate that the job should be performed on the last day of the month, irregardless of the date.
  - Use the `Months` field to select the months in which the job will execute.
- Use the fields within the `Times` box to specify the times at which the job will execute:
  - Use the `Hours` field to select the hour at which the job will execute.
  - Use the `Minutes` field to select the minute at which the job will execute.

Select the `Exceptions` tab to specify any days on which the schedule will `not` execute.

PEM scheduled job dialog exceptions tab

Use the fields on the `Exceptions` tab to specify days on which you wish the job to not execute; for example, you may wish for jobs to not execute on national holidays.

Click the Add icon (+) to add a row to the exception table, then:

- Click within the `Date` column to open a calendar selector, and select a date on which the job will not execute. Specify `<Any>` in the `Date` column to indicate that the job should not execute on any day at the time selected.
- Click within the `Time` column to open a time selector, and specify a time on which the job will not execute. Specify `<Any>` in the `Time` column to indicate that the job should not execute at any time on the day selected.

Select the `Notifications` tab to configure the email notification settings on job level:

PEM scheduled job dialog notifications tab

Use the fields on the `Notifications` tab to configure the email notification settings for a job:

- Use the `Send the notifications` field to specify when you want the email notifications to be sent.
- Use the `Email group` field to specify the email group that should receive the email notification.

When you've finished defining the schedule, you can use the `SQL` tab to review the code that will create or modify your job.

PEM scheduled job dialog SQL tab

Click the `Save` button to save the job definition, or `Cancel` to exit the job without saving. Use the `Reset` button to remove your unsaved entries from the dialog.

After saving a job, the job will be listed under the `Jobs` node of the PEM tree control of the server on which it was defined. The `Properties` tab in the PEM console will display a high-level overview of the selected job, and the Statistics tab will show the details of each run of the job. To modify an existing job or to review detailed information about a job, right-click on a job name, and select `Properties` from the context menu.

1.6 Managing a PEM Agent

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

Agent Privileges

Agent Privileges

By default, the PEM agent is installed with `root` privileges for the operating system host and superuser privileges for the database server. These privileges allow the PEM agent to invoke unrestricted probes on the monitored host and database server about system usage, retrieving and returning the information to the PEM server.

Please note that PEM functionality diminishes as the privileges of the PEM agent decrease. For complete functionality, the PEM agent should run as `root` . If the PEM agent is run under the database server's service account, PEM probes will not have complete access to the statistical information used to generate reports, and functionality will be limited to the capabilities of that account. If the PEM agent is run under another lesser-privileged account, functionality will be limited even further.

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

Note

The above-mentioned list is not comprehensive, but should provide an overview of the type of functionality that will be limited.

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

Probe	Operating System	PEM Functionality Affected
Audit Log Collection	Linux/Windows	PEM will receive empty data from the PEM database.
Server Log Collection	Linux/Windows	PEM will be unable to collect server log information.
Database Statistics	Linux/Windows	The Database/Server Analysis dashboards will contain incomplete info
Session Waits/System Waits	Linux/Windows	The Session/System Waits dashboards will contain incomplete informa
Locks Information	Linux/Windows	The Database/Server Analysis dashboards will contain incomplete info
Streaming Replication	Linux/Windows	The Streaming Replication dashboard will not display information.
Slony Replication	Linux/Windows	Slony-related charts on the Database Analysis dashboard will not disp
Tablespace Size	Linux/Windows	The Server Analysis dashboard will not display complete information.
xDB Replication	Linux/Windows	PEM will be unable to send xDB alerts and traps.

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

If the probe is querying the database with insufficient privileges, the probe may return a `permission denied` error or display the returned data in a PEM chart or graph as an empty value.

When a probe fails, an entry will be written to the log file that contains the name of the probe, the reason the probe failed, and a hint that will help you resolve the problem.

You can view probe-related errors that occurred on the server in the `Probe Log Dashboard` , or review error messages in the PEM worker log files. On Linux, the default location of the log file is:

> `/var/log/pem/worker.log`

On Windows, log information is available on the `Event Viewer` .

Agent Configuration

Agent Configuration

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

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

On Linux systems, PEM configuration options are stored in the `agent.cfg` file, located in `/usr/edb/pem/agent/etc` . The `agent.cfg` file contains the following entries:

Parameter Name	Description
pem_host	The IP address or hostname of the PEM server.
pem_port	The database server port to which the agent connects to communicate with the PEM server.
pem_agent	A unique identifier assigned to the PEM agent.
agent_ssl_key	The complete path to the PEM agent's key file.
agent_ssl_crt	The complete path to the PEM agent's certificate file.
agent_flag_dir	Used for HA support. Specifies the directory path checked for requests to take over monitorin
log_level	Log level specifies the type of event that will be written to the PEM log files.
log_location	Specifies the location of the PEM worker log file.
agent_log_location	Specifies the location of the PEM agent log file.
long_wait	The maximum length of time (in seconds) that the PEM agent will wait before attempting to co
short_wait	The minimum length of time (in seconds) that the PEM agent will wait before checking which p
alert_threads	The number of alert threads to be spawned by the agent.
enable_smtp	When set to true, the SMTP email feature is enabled.
enable_snmp	When set to true, the SNMP trap feature is enabled.
enable_nagios	When set to true, Nagios alerting is enabled.
connect_timeout	The max time in seconds (a decimal integer string) that the agent will wait for a connection.
allow_server_restart	If set to TRUE, the agent can restart the database server that it monitors. Some PEM features
max_connections	The maximum number of probe connections used by the connection throttler.
connection_lifetime	Use ConnectionLifetime (or connection_lifetime) to specify the minimum number of seconds a
allow_batch_probes	If set to TRUE, the user will be able to create batch probes using the custom probes feature.
heartbeat_connection	When set to TRUE, a dedicated connection is used for sending the heartbeats.
batch_script_dir	Provide the path where script file (for alerting) will be stored.
connection_custom_setup	Use to provide SQL code that will be invoked when a new connection with a monitored server
ca_file	Provide the path where the CA certificate resides.

On 64 bit Windows systems, PEM registry entries are located in:

> `HKEY_LOCAL_MACHINE\Software\Wow6432Node\EnterpriseDB\PEM\agent` .

The registry contains the following entries:

Agent Properties

Agent Properties

The `PEM Agent Properties` dialog provides information about the PEM agent from which the dialog was opened; to open the dialog, right-click on an agent name in the PEM client tree control, and select `Properties` from the context menu.

The PEM Agent Properties dialog

Use fields on the PEM Agent properties dialog to review or modify information about the PEM agent:

- The `Description` field displays a modifiable description of the PEM agent. This description is displayed in the tree control of the PEM client.
- You can use groups to organize your servers and agents in the PEM client tree control. Use the `Group` drop-down listbox to select the group in which the agent will be displayed.
- Use the `Team` field to specify the name of the group role that should be able to access servers monitored by the agent; the servers monitored by this agent will be displayed in the PEM client tree control to connected team members. Please note that this is a convenience feature. The Team field does not provide true isolation, and should not be used for security purposes.
- The `Heartbeat interval` fields display the length of time that will elapse between reports from the PEM agent to the PEM server. Use the selectors next to the `Minutes` or `Seconds` fields to modify the interval.

1.7 Conclusion

Postgres Enterprise Manager Getting Started Guide

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

- EDB designs, establishes coding best practices, reviews, and verifies input validation for the logon UI for Postgres Enterprise Manager 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.

2.0 PEM Installation Guide

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

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

For information about the platforms and versions supported by PEM, visit the EnterpriseDB website at:

> https://www.enterprisedb.com/services-support/edb-supported-products-and-platforms#pem

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

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

2.1 What's New

The following features have been added to create Postgres Enterprise Manager 7.13:

- Core Usage Report: The Core Usage report provides you with metrics such as total number of cores used by a EDB Postgres Advanced Server or Postgres database that is being managed by PEM. This report can help you understand if you are in compliance with core-based licensing guidelines.
- SMMP v3 support: SNMP v3 support enables the PEM Server and PEM Agent to use SNMP Version 3 for secure communication by authenticating and encrypting data packets over the network.
- Schema Diff tool (Beta): The Schema Diff tool allows you to compare two database schema objects and synchronize the two schemas. The tool lists the differences and also generates the synchronization scripts for the two schemas.
- Auto logout inactive users: An inactive user session is automatically logged out after a certain period of inactivity. This timeout interval can be configured in the `config.py` file. This feature improves security by reducing the unintended access to the database server.
- Accessibility Improvements: Enables screen-readers to read labels and descriptions of non-text elements, to identify the alert errors, and to read relationship attributes in nested elements.
- Other features include:
  - Enhancements to the System Configuration Report.
  - Support for a multi-level partitioned table.

2.2 Postgres Enterprise Manager - Overview

Postgres Enterprise Manager (PEM) consists of components that provide the management and analytical features of PEM:

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

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

  The PEM worker log file contains information related to PEM worker activity (probe activities, heartbeat responses, etc.), and is stored in `/var/log/pem/worker.log` .
- **SQL Profiler plugin:** This plugin to the Postgres server is used to generate the monitoring data used by the SQL Profiler tool. Installation of the SQL Profiler plugin is optional, but the plugin must be installed into each instance of Postgres you wish to profile. The SQL Profiler may be used with any supported version of an EnterpriseDB distribution of a PostgreSQL server or an Advanced Server (not just those managed through the PEM server). See Installing the SQL Profiler Plugin for details and supported versions.

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

A typical PEM installation

Hardware Prerequisites

Hardware Prerequisites

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

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

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

Software Prerequisites

Software Pre-Requisites

Modifying the pg_hba.conf File

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

For information about modifying the `pg_hba.conf` file, see the *PEM Administrator's Guide* available at:

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

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

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

Firewall Restrictions

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

PEM Agent Installation- Prerequisites for RHEL or CentOS 7.x

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

```
yum install epel-release
```

Windows Permissions

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

Supported Locales

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

Windows Server IE Security

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

2.3.0 Installing Postgres Enterprise Manager

installing PEM

You can use a graphical installer, StackBuilder Plus, or an RPM package to add the Postgres Enterprise Manager server or agent to a host. Installers are available on the EnterpriseDB website.

The PEM server graphical installer for Windows installs and configures the PEM server, a PEM agent, and the software required to connect to the PEM web interface with your choice of browser. For detailed information about using the PEM server graphical installer, see Installing the PEM Server on Windows.

You can use an RPM package to install the PEM server on a Linux host; for detailed information about using the installer, see Installing the PEM Server on Linux.

The PEM agent graphical installer for Windows installs and registers the PEM agent. The PEM agent that is installed with the PEM server is capable of monitoring multiple servers that reside on the same host, or on remote hosts. Please note that the PEM functionality on servers monitored by a remote agent may be limited.

For more information about using the PEM agent graphical installer, see Installing a PEM Agent on Windows.

You can use an RPM package to install the PEM agent on Linux; for more information about the PEM agent RPM package, see Installing a PEM Agent on Linux.

2.3.1 Installing the PEM Server on Windows

installing PEM server on Windows

installing PEM server through graphical installer

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

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

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

  For more information about using an existing Postgres server to host the PEM server backend database, see Installing the PEM Server on an Existing Postgres Server section.

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

  This is the simplest PEM server installation option.

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

Before installing the PEM server, you must decide if you wish to run PostgreSQL and PEM-HTTPD on the same host or on separate hosts. If you intend to run the PostgreSQL database server and PEM-HTTPD on different hosts, then you must run the PEM server installer twice – once on each host, as detailed in Installing the PEM Server and PEM-HTTPD on Separate Hosts section.

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

The PEM web interface

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

installing PEM Server and PEM-HTTPD on same Host

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

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

To invoke the PEM server installer on a Windows system, right click the installer icon and select `Run as Administrator` . The installer displays a `Welcome` dialog.

The PEM Server Installer's Welcome dialog

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

The License Agreement

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

Specify an installation directory

Use the `Installation Directory` dialog to specify the location of the PEM server:

- By default, the PEM server is installed in `C:\Program Files\edb\pem` on Windows. Accept the default location, or use the `Installation Directory` button to open a browser dialog and select the directory in which the PEM server will be installed.
- Use the `Show advanced options` check box to instruct the installer to open the `Advanced options` dialog.
- Use the `Advanced options` dialog when installing the Postgres database server and the PEM-HTTPD on different hosts, or if you wish the PEM server to reside on an existing Postgres server installation.

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

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

Note

By default EDB Language Pack is installed in `C:\edb\languagepack\v1` .

The installer checks for pre-requisites

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

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

Enter the name and password of the PostgreSQL superuser

The information provided on the `Database Server Installation Details` dialog enables the installer to connect to the PostgreSQL server. Provide the `User name` and `Password` of a database superuser. After supplying the requested information, click `Next` to continue to the `pemAgent Service Account` dialog.

pemAgent Service Account Password

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

Supply the network address from which the agent will connect

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

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

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

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

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

Provide a Description for the agent that resides on the server

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

The installation is ready to begin

The wizard is now ready to install the PEM server.

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

The installation in progress

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

A popup confirms the installation details

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

The PEM server installation is complete

Installing the PEM Server and PEM-HTTPD on Separate Hosts

installing PEM Server and PEM-HTTPD on separate Hosts

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

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

To invoke the PEM server installer on a Windows system, right click the installer icon and select `Run as Administrator` . The installer displays a `Welcome` dialog.

The PEM Server Installer's Welcome dialog

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

The License Agreement

Carefully review the license agreement before highlighting the appropriate radio button and accepting the agreement; click `Next` to continue.

The Installation Directory dialog

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

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

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

The Advanced Options dialog

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

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

Note

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

This option invokes the installation steps documented in Installing Web Services section.

- Select `Database` to use an existing Postgres server (version 9.4 or greater), or to install only the database server that is distributed with the PEM server installer. This option invokes the installation steps documented in Specifying a Database Host.

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

Specifying a Database Host

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

The Advanced options dialog

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

Selecting a database server

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

- Select a host from existing Postgres installations that reside on the current host.

Note

You may be required to add the `sslutils` package to your installation.

- Select the `PostgreSQL x` option to install the Postgres server that is distributed with the PEM server installer where `x` is the PostgreSQL database server version. If you decide to use the version of PostgreSQL that is bundled with the PEM server installer, the EnterpriseDB one-click PostgreSQL installer will open and walk you through the installation.

- Select `Other Database Server` to specify connection information for a Postgres server that was not installed using a one-click graphical installer from EnterpriseDB. For information about the software pre-requisites for the PEM server database host, please see Preparing the Postgres Server section.

Click `Next` to continue.

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

The Database Server Installation Details dialog

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

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

Click `Next` to continue.

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

Supply the network address from which the agent will connect

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

Note

You can specify additional network addresses by manually adding entries to the `pg_hba.conf` file on the PostgreSQL server.

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

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

Provide a Description for the agent

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

The installation is ready to begin

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

The installation in progress

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

Click OK to confirm that the pem database has been created

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

The PEM server installation is complete

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

Installing Web Services

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

Selecting the Web Services option on the Advanced options dialog

Note

The current host may not be the host of the PEM backing database.

Before selecting this option, you must have:

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

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

If the installer does not locate the components, the installer will inform you that one or more dependencies are missing.

The installer has detected missing dependencies

Click `Next` to instruct the server to install LanguagePack

The installer has detected missing dependencies

After installing language pack, the installer will invoke the PEM-HTTPD setup wizard.

The PEM-HTTPD installation wizard

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

The Database Server Installation Details dialog

Use the fields on the `Database Server Installation Details` dialog to provide connection information for the Postgres installation that is hosting the PEM server installation:

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

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

Installing the PEM Server on an Existing Postgres Server

Installing the PEM Server on an Existing Postgres Server

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

This section provides information about configuring an existing Postgres server for a PEM server installation.

Note

The steps that follow should be considered guidelines only; the actual steps required to configure your Postgres installation will vary depending on the configuration of your Postgres server.

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

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

Preparing the Postgres Server

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

- Ensure that the Postgres server contains an installation of the `sslutils` extension. For more information, see Installing the sslutils Extension section.
- Create a service script (on Linux) or register the server with the Windows service manager. For more information, see Creating a Service Script or Registering the Service section.

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

Installing the sslutils Extension   Installing the sslutils Extension

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

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

On Linux

Installing the sslutils Extension on Linux

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

1. Use yum to install gcc:

   ```
   yum install gcc
   ```

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

   `export PATH=$PATH:/usr/pgsql-x/bin` where `x` is the version of PostgreSQL database server

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

   ```
   yum install openssl-devel
   ```

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

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

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

   ```
   CREATE EXTENSION sslutils
   ```

On Windows

Installing the sslutils Extension on Windows

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

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

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

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

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

   ```
   SET USE_PGXS=1
   ```

   ```
   SET GETTEXTPATH=<path_to_gettext>
   ```

   ```
   SET OPENSSLPATH=<path_to_openssl>
   ```

   ```
   SET PGPATH=<path_to_pg_installation_dir>
   ```

   ```
   SET ARCH=x86
   ```

   ```
   REM Set ARCH x64 for 64 bit
   ```

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

   Where:

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

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

   `path_to_pg_installation_dir` specifies the location of the Postgres installation.

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

   ```
   sslutils--1.1.sql
   ```

   ```
   sslutils--unpackaged--1.1.sql
   ```

   ```
   sslutils--pemagent.sql.in
   ```

   ```
   sslutils.dll
   ```

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

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

   ```
   COPY sslutils.dll "%PGPATH%\lib\"
   ```

Creating a Service Script or Registering the Service Creating a Service Script or Registering the Service

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

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

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

Writing a Linux Service Script

Writing a Linux Service Script

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

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

Registering a Service on Windows

Registering a Service on Windows

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

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

> Where:
>
> `service name` specifies the name of the Postgres cluster.
>
> `user_name` specifies the name of an operating system user with sufficient privileges to access the Postgres installation directory and start the Postgres service.
>
> `password` specifies the operating system password associated with the user.
>
> `data_directory` specifies the location of the Postgres data directory.

For more information about using the `pg_ctl` command and the available command options, see the

> Postgres core documentation

Invoking the PEM Server Installer

Invoking the PEM Server Installer

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

```
./pem_server-7.<x>.<x>-<x>-<platform>
```

Where *x* is the major and minor versions of PEM and platform is the platform.

The installer displays a `Welcome` dialog.

The PEM server Installer's Welcome Dialog

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

The License Agreement

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

The Installation Directory dialog

Use the `Installation Directory` dialog to specify the location of the PEM server and access the `Advanced options` dialog:

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

Click `Next` to continue.

The Advanced Options dialog

Use the radio buttons on the `Advanced options` dialog to specify an installation type. Select:

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

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

- `Web Services` if only the PEM-HTTPD server will reside on the current host. See Installing Web Services section for more information about invoking this option.

- `Database` if you are installing only the PEM server (and creating the `pem` backend database) on the current host. This option is valid if you are using an existing Postgres server to host the PEM server, or using the PEM server installer to install the PostgreSQL server on which PEM will reside.

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

The Database Server Selection dialog

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

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

Note

The selected database server must include an installation of the `sslutils` contrib module, and have a startup script (on Linux) or a registered service (on Windows).

For information about Preparing the Postgres Server, please see this section.

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

If you selected `Database` on the `Advanced options` dialog, the `Database Server Installation Details` dialog opens.

The Database Server Installation Details dialog

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

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

Note

For information about preparing a service script, please see this section

Click `Next` to continue.

Supply the network address from which the agent will connect

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

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

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

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

Provide a Description for the agent that resides

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

The installation is ready to begin

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

The installation in progress

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

Click OK to confirm that the pem database has been created

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

The PEM server installation is complete

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

- Modify the `pg_hba.conf` file on the Postgres server host to allow connections between the hosts.
- Invoke the PEM server installer on the host of the PEM-HTTPD server. See Installing Web Services section for more information about installing PEM-HTTPD.

Invoking the Server Installer from Command Line

invoking server installers from command line

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

- Include the `--mode unattended` option when invoking the installer to perform an installation without additional user input.

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

Invoking the PEM Server Installer in Unattended Mode

invoking PEM server installer in unattended mode

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

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

- PostgreSQL
- pem-httpd
- Language Pack

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

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

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

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

postgresql-<version>-windows-x64.exe --mode unattended

Then, you can invoke the PEM server installer:

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

Where:

`registered_edb_user` specifies the name of a registered EnterpriseDB user. To register, visit the EDB website

`edb_user_password` specifies the password associated with the EDB user account.

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

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

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

Note

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

EnterpriseDB is the leading provider of value-added products and services for the Postgres community. Please visit our website at www.enterprisedb.com

---

2.3.2 Installing the PEM Server on Linux

installing PEM server on Linux

When installing a PEM server on a RHEL, CentOS, SLES, Debian, or Ubuntu host, you must ensure the following:

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

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

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

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

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

3. If you are using a PostgreSQL database, you must also install the `hstore contrib` module; for more information, visit:

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

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

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

   ```
   firewall-cmd --reload
   ```

Additional Prerequisites for RHEL or CentOS HOST

In addition to the above listed prerequisites, the following prerequisites are applicable if you are using a RHEL or CentOS host:

1. You must install the `epel-release` package on the host 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`).

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

2. You must also enable the `rhel-<x>-server-extras-rpms` repository, where `x` specifies the version of the RHEL on the host.

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

   1. Stop the default httpd server using the command:

      ```
      service httpd stop
      ```

   2. Run the following commands to install `rh-python36` :

      ```
      yum install centos-release-scl
      ```

      ```
      yum install rh-python36
      ```

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

After receiving your repository credentials you can:

1. Create the repository configuration file.
2. Modify the file, providing your user name and password.
3. Install `edb-pem` on RHEL or CentOS host.

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
```

Installing the PEM Server on a CentOS or RHEL Host

You can use yum to install the PEM server:

```
yum install edb-pem
```

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

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

If you want to install PEM server on a machine that is in isolated network, you must first create PEM repository on that machine. For more information about creating PEM repository on an isolated network, see Creating a PEM repository in an Isolated Network.

Installing the PEM Server on a Debian or Ubuntu Host

To install PEM on a Debian or Ubuntu host, you must have credentials that allow access to the EnterpriseDB repository. To request credentials for the repository, contact EnterpriseDB .

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

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

   ```
   sudo su -
   ```

2. Configure the EnterpriseDB repository:

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

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

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

4. Add the EBD signing key:

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

5. Update the repository metadata:

   ```
   apt-get update
   ```

6. Use the following command to install the Debian package for the PEM server:

```
apt-get install edb-pem
```

Installing PEM Server on a SLES Host

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

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

- SLES packages are available from:
  https://zypp.enterprisedb.com

Before installing PEM, you must install prerequisite packages. Invoke the following commands, replacing *sp_no* with the service pack that you are using (i.e. SP4):

```
SUSEConnect -p sle-module-legacy/12/x86_64
```

```
SUSEConnect -p sle-sdk/12/x86_64
```

```
zypper addrepo  https://download.opensuse.org/repositories/Apache:Modules/SLE_12_<sp_no>/
```

```
zypper addrepo http://download.opensuse.org/repositories/Cloud:/OpenStack:/Newton:/cisco-
```

```
zypper refresh
```

```
zypper install edb-pem
```

Configuring the PEM Server

Before configuring the PEM server, ensure that the `sslutils` extension is installed for your backing database.

- For an Advanced Server backing database, `sslutils` extension is by default installed along with Advanced Server.
- If you are using a PostgreSQL backing database, ensure you have access to the PostgreSQL community repository, and use the command:

  ```
  yum install sslutils_<x> postgresql<X>-contrib
  ```

Where, `x` is the server version.

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

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

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

Option	Description
`-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
`-dbi`	The directory for the database server installation. For example, `/usr/edb/as10` for Advanced Server or `/us`
`-ds`	The unit file name of the PEM database server. For Advanced Server, the default file name is `edb-as-10` ; for
`-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` .

Option	Description
`-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 configu

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

1. `Web Services and Database` -Select this option if the web server and database both reside on the same host as the PEM server.

2. `Web Services` -Select this option if the web server resides on a different host than the PEM server.

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

Note

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

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

To view script-related help, use the command:

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

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

```
service httpd24-httpd restart
```

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

https://<ip_address_of_PEM_server>:8443/pem

2.3.3 Creating a PEM Repository in an Isolated Network

You can create a local repository to act as a host for PEM RPM packages if the server on which you wish to install 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. Download all the dependencies required for PEM on a machine with internet access using the following commands:

```
yum install yum-plugin-downloadonly

mkdir /tmp/<pem_dir>

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

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:

```
yum install createrepo

createrepo /tmp/<pem_dir>

createrepo /tmp/<epel_dir>
```

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

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

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

```
[epelrepo]
name=epel Repository
baseurl=file:///tmp/<epel_dir>/
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

2.3.4 Installing a PEM Agent on Windows

installing agent on windows

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

```
pem_agent-7.<x>.<x>-<x>-platform.exe
```

The `Setup…` page opens, welcoming you to the PEM Agent installer.

The PEM Agent Installer's Welcome dialog

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

The PEM License Agreement

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

Specify an Installation Directory

By default, the PEM agent is installed in the `C:\Program Files (x86)\edb\pem` directory.You can accept the default installation directory, or modify the contents of the `Installation Directory` field, specifying an alternate installation directory for the PEM agent.

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

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

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

Provide PEM server installation details

Enter the connection details for the PEM server on the `PEM server installation details` dialog:

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

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

Note

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

pemAgent Service Account password

Provide the password for the edb account under which the pemAgent service will run. The agent certificate and key files will be created in `C:\Users\edb\App` Provide the password for the edb account under which the pemAgent service will run. The agent certificate and key files will be created in `C:\Users\edb\AppData\Roaming\pem` directory. Click `Next` to continue to `Agent Details` dialog.

Specify a descriptive name for the PEM agent

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

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

If you checked the `Show advanced options` checkbox, the `Advanced options` dialog opens.

Provide the certificate path

By default, the PEM agent installer places the certificate in the `C:\Program Files (x86)\edb\pem` directory. Specify an alternate path for the certificate or accept the default and click `Next`.

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

The PEM Agent installation is ready to begin

Click `Next` on the `Ready to Install` dialog to instruct the installer to copy files to the system and register the agent on the PEM server.

Progress bars mark the installation's progress

The PEM agent installer displays progress bars to mark the PEM agent's installation progress.

The PEM Agent installation is complete

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

Invoking the Agent Installer from the Command Line

invoking agent Installers from command line

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

- Include the `--mode unattended` option when invoking the installer to perform an installation without additional user input.

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

Invoking the PEM Agent Installer in Unattended Mode

Invoking the PEM Agent Installer in Unattended Mode

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

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

- install the PEM server; the `pg_hba.conf` file of the PEM server must allow connections from the host of the PEM agent.
- disable SE Linux on the host of the PEM agent; for more information, see section Software Prerequisites.
- ensure that the monitored Postgres database has SSL enabled, and is accepting connections.

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

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

Where:

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

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

`pguser_password` specifies the password associated with the PEM database superuser.

`agent_name` specifies a descriptive name for the PEM agent.

```
EnterpriseDB is the leading provider of value-added products and services for the Postgres communi
Please visit our website at www.enterprisedb.com.
```

Note

When configuring a shell/batch script run by a Windows agent that has PEM 7.11 or later version installed, the `AllowBatchJobSteps` parameter must be set to `True` in the `agent.cfg` file. The pemagent will not execute any batch/shell script by default.

2.3.5 Installing a PEM Agent on Linux

installing agent on Linux

Installing a PEM agent on a CentOS or RHEL host

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

- Install the `epel-release` package on the host 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`).

- 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

After receiving your repository credentials you can:

1. Create the repository configuration file.
2. Modify the file, providing your user name and password.
3. Install `edb-pem-agent` .

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
```

Installing PEM Agent

After saving your changes to the configuration file, you can use the yum install command to install `edb-pem-agent` . For example, the following command installs PEM Agent:

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

When the installation is complete, `yum` will display a list of the installed packages and dependencies.

Using an RPM package to install the PEM agent

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

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

Installing a PEM Agent on a Debian or Ubuntu Host

To install PEM on a Debian or Ubuntu host, you must have credentials that allow access to the EnterpriseDB repository. To request credentials for the repository, contact EnterpriseDB .

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

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

   ```
   sudo su -
   ```

2. Configure the EnterpriseDB repository:

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

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

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

4. Add the EBD signing key:

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

5. Update the repository metadata:

   ```
   apt-get update
   ```

6. Use the following command to install the Debian package for the PEM agent:

   ```
   apt-get install edb-pem-agent
   ```

Installing a PEM Agent on a SLES Host

Installing PEM Agent on a SLES Host

For detailed information about installing Advanced Server and supporting components on a SLES host, please consult the

EDB Postgres Advanced Server Installation Guide

SLES packages are available from:

https://zypp.enterprisedb.com

Before installing PEM, you must install prerequisite packages. Use the following commands replacing *sp_no* with the service pack that you are using (i.e. SP2 or SP3):

```
SUSEConnect -p sle-module-legacy/12/x86_64

SUSEConnect -p sle-sdk/12/x86_64

zypper addrepo https://download.opensuse.org/repositories/Apache:Modules/SLE_12_<sp_no>/Apache:

zypper addrepo http://download.opensuse.org/repositories/Cloud:/OpenStack:/Newton:/cisco-
apic:/2.3.1/SLE_12_<sp_no>/ pem_opensuse_boost

zypper refresh

zypper install edb-pem-agent
```

Registering an Agent

Registering an Agent

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

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

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

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

Option	Description
`--pem-server`	Specifies the IP address of the PEM backend database server. This parameter i
`--pem-port`	Specifies the port of the PEM backend database server. The default value is 543
`--pem-user`	Specifies the name of the Database user (having superuser privileges) of the PE
`--pem-agent-user`	Specifies the agent user to connect the PEM server backend database server.
`--cert-path`	Specifies the complete path to the directory in which certificates will be created.
`--config-dir`	Specifies the directory path where configuration file can be found. The default is
`--display-name`	Specifies a user-friendly name for the agent that will be displayed in the PEM Br
`--force-registration`	Include the force_registration clause to instruct the PEM server to register the ag
`--group`	The name of the group in which the agent will be displayed.
`--team`	The name of the database role, on the PEM backend database server, that shou
`--owner`	The name of the database user, on the PEM backend database server, who will
`--allow_server_restart`	Enable the allow-server_restart parameter to allow PEM to restart the monitored
`--allow-batch-probes`	Enable the allow-batch-probes parameter to allow PEM to run batch probes on t
`--batch-script-user`	Specifies the operating system user that should be used for executing the batch
`--enable-heartbeat-connection`	Enable the enable-heartbeat-connection parameter to create a dedicated heartb
`--enable-smtp`	Enable the enable-smtp parameter to allow the PEM agent to send the email on
`--enable-snmp`	Enable the enable-snmp parameter to allow the PEM agent to send the SNMP t
`-o`	Specify if you want to override the configuration file options.

If you want to use any PEM feature for which database server restart is required by the pemagent such as Audit Manager, Log Manager, or Tuning Wizard, then you must set the value for `allow_server_restart` as `true` in the `agent.cfg` file.

Note

When configuring a shell/batch script run by a PEM agent that has PEM 7.11 or later version installed, the user for the `batch_script_user parameter` must be specified. 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` .

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

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

Failure to provide the password will result in a password authentication error; you will be prompted for any other required but omitted information. When the registration is complete, the server will confirm that the agent has been successfully registered.

Setting PEM Agent Configuration Parameters

The PEM agent RPM installer creates a sample configuration file named `agent.cfg.sample` in the

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

```
heartbeat_connection = true
```

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

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

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

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

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

```
systemctl start pemagent
```

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

For information about using the `pemworker` utility to register a server, please see the PEM Administrator's Guide

---

2.4 The PEM Web Interface

The PEM Web Interface

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

After installing and configuring PEM, you can use your browser to access the PEM web interface. Open your browser, and navigate to:

```
https://<ip_address_of_PEM_host>:8443/pem
```

Where *ip_address_of_PEM_host* specifies the IP address of the host of the PEM server. The `Postgres Enterprise Man` window opens:

The PEM Web Login page

Use the fields on the `Postgres Enterprise Manager Login` window to authenticate yourself with the PEM server:

- Provide the name of a `pem` database user in the `Username` field. For the first user connecting, this will be the name provided when installing the PEM server.
- Provide the password associated with the user in the `Password` field.

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

The PEM Web interface

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

A server may be managed or unmanaged:

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

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

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

- The `PEM Administrator's Guide` contains information about registering and managing servers, agents, and users.
- The `PEM Enterprise Features Guide` contains information about using the tools and wizards that are part of the web interface.
- The `PEM Agent User Guide` contains helpful information about managing your PEM agents.
- The `PEM Upgrade and Migration Guide` contains information about upgrading to PEM 7.8 from a previous version.
- The `PEM PgBouncer Configuration Guide` contains information about using PgBouncer with your PEM installation.
- The `PEM EDB Ark Management Guide` contains information about using PEM to manage cloud installations of Advanced Server and PostgreSQL.

2.5 Installing the SQL Profiler Plugin

Installing the SQL Profiler Plugin

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

SQL Profiler is officially supported only on the EnterpriseDB distributions of PostgreSQL version 9.4 or above and Advanced Server version 9.4 or above. The plugin is distributed via StackBuilder, or is available from the EnterpriseDB website

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

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

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

You can use the graphical installer to install any version of SQL Profiler on Windows platform.

On Linux, if you have installed your database server through graphical installer then you must use the graphical installer to install the SQL Profiler. If you have installed your database server using the RPM package, then you must use the RPM package to install the SQL Profiler.

Installing SQL Profiler using Graphical Installer

Installing SQL Profiler using graphical installer

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

```
postgresql<pg_version>-sqlprofiler-<sql_profiler_version>
```

Where, `pg_version` is the version of your postgres and `sql_profiler_version` is the version of SQL Profiler.

The SQL Profiler installer welcomes you to the Setup Wizard.

The SQL Profiler Installer Welcome dialog

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

The SQL Profiler License Agreement

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

The PostgreSQL Installation Directory

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

Ready to Install

The wizard is now ready to install the SQL Profiler plugin. Click `Next` to continue.

Installing the SQL Profiler plugin

The SQL Profiler plugin installer displays progress bars as it copies files to your system.

The SQL Profiler installation is complete

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

Installing SQL Profiler using RPMs

Installing SQL Profiler using RPMs

You can install SQL Profiler using rpm on RHEL or Centos 6 or 7, using yum command as root user:

```
yum install postgresql<pg_version>-sqlprofiler-<sql_profiler_version>
```

Where, `pg_version` is the version of your postgres and `sql_profiler_version` is the version of SQL Profiler.

Configuring SQL Profiler

Configuring SQL Profiler

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

Use the following steps to enable the plugin:

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

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

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

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

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

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

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

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

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

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

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

Creating a new trace

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

2.6 Upgrading and Uninstalling PEM Components

Upgrading and Uninstalling PEM Components

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

Upgrading Postgres Enterprise Manager on Windows

Upgrading Postgres Enterprise Manager on windows

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

You should invoke the PEM installers in the following order:

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

Upgrading Postgres Enterprise Manager on Linux

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

Upgrading SQL Profiler on a Linux Host

Upgrading SQL Profiler on a Linux Host

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

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

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

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

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

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

  Where *x.x* specifies the installed Postgres version.

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

Upgrading SQL Profiler on a Windows Host

Upgrading SQL Profiler on a Windows Host

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

After stopping the Postgres Server:

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

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

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

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

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

     Where *x* specifies the installed Postgres version.

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

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

Troubleshooting

Upgrading SQL Profiler on a Windows Host - Troubleshooting

Upgrading SQL Profiler on a Windows Host

After upgrading to a newer version of SQL Profiler, if you encounter the following error:

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

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

Uninstalling Postgres Enterprise Manager

Uninstalling Postgres Enterprise Manager

The process of uninstalling the PEM server or agent is platform-specific. The name of the package for PEM server is `edb-pem-server` and for PEM agent is `edb-pem-agent` .

If you uninstall the PEM server package from a host, the PEM agent package installed on the same host doesn't get uninstalled. But if you uninstall the PEM agent package, then the PEM server package installed on the same host also gets uninstalled.

Uninstalling PEM from Windows hosts

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.

You can also invoke the uninstaller that resides at the following location:

```
C:\\Program Files\edb\pem\server
```

Uninstalling PEM from CentOS or RHEL hosts

You can use variations of the `rpm` , `yum remove` , or `yum erase` commands to remove the installed packages. Note that removing a package does not damage the PEM data directory.

- Include the `-e` option when invoking the rpm command to remove an installed package; the command syntax is:

  ```
  rpm -e <package_name>
  ```

- You can use the `yum remove` command to remove the pem server or agent package installed by yum. To remove a package, open a terminal window, assume superuser privileges, and enter the command:

  ```
  yum remove <package_name>
  ```

- You can use the `yum erase` command to remove the pem server or agent package along with the `edb-pem` and `edb-pem-docs` dependencies. To remove a package, open a terminal window, assume superuser privileges, and enter the command:

  ```
  yum erase <package_name>
  ```

Where *package_name* is the name of the package that you would like to remove.

Uninstalling PEM from Debian or Ubuntu hosts

You can use `apt-get remove` or `apt-get purge` command to uninstall the PEM server or agent package from a Debian or Ubuntu host:

- To uninstall PEM server or agent from a Debian or Ubuntu host without impacting the configuration files and data directories, invoke the following command:

  ```
  apt-get remove <package_name>
  ```

- To uninstall PEM server or agent along with the configuration files and data directory, invoke the following command:

  ```
  apt-get purge <package_name>
  ```

Where *package_name* is the name of the package that you would like to remove.

Uninstalling PEM from SLES hosts

To uninstall PEM server or agent from a SLES host, invoke the following command:

```
zypper remove <package_name>
```

Where *package_name* is the name of the package that you would like to remove.

2.7 Reference - Linux Service Script

Reference - Linux Service Script

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

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

```
| #!/bin/bash
| # chkconfig: 2345 85 15
| # description: Starts and stops the PostgreSQL/Postgres Plus Advanced
Server database server
| # PostgreSQL/Postgres Plus Advanced Server Service script template for
Linux
| # Please modify the values accordingly
| DB_DESC="Database Server - PostgreSQL 9.5"
| DB_INSTALL_DIR=/opt/PostgreSQL/9.5
| DB_BIN_DIR=${DB_INSTALL_DIR}/bin
| DB_LIB_DIR=${DB_INSTALL_DIR}/lib
| DB_DATA_DIR=${DB_INSTALL_DIR}/data
| DB_HBA_FILE=${DB_DATA_DIR}/pg_hba.conf
| DB_CONF_FILE=${DB_DATA_DIR}/postgresql.conf
| DB_PID_FILE=${DB_DATA_DIR}/postmaster.pid
| DB_STARTUP_LOG=${DB_DATA_DIR}/pg_log/startup.log
| DB_SERVICE_USER=postgres
| _die()
| {
|  echo ""
|  echo "FATAL ERROR: $*"
|  echo ""
|  exit 1
| }
| if [ `id -u` != 0 ]; then
|    _die "You must run this script as the root."
| fi

| # Source function library.
| if [ -f /etc/rc.d/functions ];
| then
|    . /etc/init.d/functions
| fi
start()
| {
|  STARTDBSERVER=0
|  if [ -e "${DB_PID_FILE}" ]
|  then
|    PIDOFDB=`head -n 1 "${DB_PID_FILE}"`
|    PIDALIVEDB=""
|    if [ -n "${DB_PID_FILE}" ]; then
|       PIDALIVEDB=`ps -p "${PIDOFDB}" | grep
"${PIDOFDB}"`
|     fi
|     if [ -n "${PIDALIVEDB}" ]
|     then
|        echo "The '${DB_DESC}' is already running.
             PID(${PIDALIVEDB})."
|                    exit
|            else
|                    STARTDBSERVER=1
|            fi
|             else
|            STARTDBSERVER=1
|              fi
| if [ "${STARTDBSERVER}" != 0 ]
| then
| echo "Starting ${DB_DESC}..."
| su - "${DB_SERVICE_USER}" -c
"LD_LIBRARY_PATH=\"${DB_LIB_DIR}:\$LD_LIBRARY_PATH\"
\"${DB_BIN_DIR}/pg_ctl\" -w start -D \"${DB_DATA_DIR}\" -l
\"${DB_STARTUP_LOG}\" -o \"${DB_STARTUP_OPTIONS}\""
| if [ $? -eq 0 ];
| then
| echo "${DB_DESC} started successfully."
| exit 0
| else
| echo "${DB_DESC} did not start in a timely fashion, please see
'${DB_STARTUP_LOG}' for details."
| exit 1
| fi
| fi
| }
| stop()
| {
| if [ -e "${DB_PID_FILE}" ]
| then
| PIDOFDB=`head -n 1 "${DB_PID_FILE}"`
| PIDALIVEDB=""
| if [ -n "${DB_PID_FILE}" ]; then
| PIDALIVEDB=`ps -p "${PIDOFDB}" | grep "${PIDOFDB}"`
| fi
| if [ -n "${PIDALIVEDB}" ]
| then
| echo "Stopping ${DB_DESC}..."
| su - "${DB_SERVICE_USER}" -c
"LD_LIBRARY_PATH=\"${DB_LIB_DIR}:\$LD_LIBRARY_PATH\"
\"${DB_BIN_DIR}/pg_ctl\" stop -m fast -D \"${DB_DATA_DIR}\"
-l \"${DB_STARTUP_LOG}\" -o \"${DB_STARTUP_OPTIONS}\""
| else
| echo "The '${DB_DESC}' is not running."
| fi
| else
| echo "The '${DB_DESC}' is not running."
| fi
| }
| reload()
| {
| echo "Reloading '${DB_DESC}'..."
| su - "${DB_SERVICE_USER}" -c
"LD_LIBRARY_PATH=\"${DB_LIB_DIR}:\$LD_LIBRARY_PATH\"
\"${DB_BIN_DIR}/pg_ctl\" reload -D \"${DB_DATA_DIR}\" -l
\"${DB_STARTUP_LOG}\" -o \"${DB_STARTUP_OPTIONS}\""
| }
| restart()
| {
| echo "Restarting '${DB_DESC}'..."
| su - "${DB_SERVICE_USER}" -c
"LD_LIBRARY_PATH=\"${DB_LIB_DIR}:\$LD_LIBRARY_PATH\"
\"${DB_BIN_DIR}/pg_ctl\" restart -m fast -w -D
\"${DB_DATA_DIR}\" -l \"${DB_STARTUP_LOG}\" -o
\"${DB_STARTUP_OPTIONS}\""
| if [ $? -eq 0 ];
| then
| echo "'${DB_DESC}' restarted successfully."
| exit 0
| else
| echo "'${DB_DESC}' did not start in a timely fashion, please see
'${DB_STARTUP_LOG}' for details."
| exit 1
| fi
| }
| _die_incomplete_requirement()
| {
| echo "One or more required configuration variables are not set:"
| _die $*
| }
| _validate_script()
| {
| if [ -z "${DB_INSTALL_DIR}" ]; then
| _die_incomplete_requirement "Missing installation directory";
| fi
| if [ ! -d "${DB_INSTALL_DIR}" ]; then
| _die_incomplete_requirement "The specified - '${DB_INSTALL_DIR}'
is not a valid installation directory. It is not present on the
system.";
| fi
| if [ -z "${DB_BIN_DIR}" ]; then
DB_BIN_DIR=${DB_INSTALL_DIR}/bin; fi
| if [ ! -d "${DB_BIN_DIR}" ]; then
| _die_incomplete_requirement "The specified - '${DB_BIN_DIR}' is
not a valid bin directory. It is not present on the system.";
| fi
| if [ ! -f "${DB_BIN_DIR}/pg_config" -o ! -f
"${DB_BIN_DIR}/pg_ctl" ]; then
| _die_incomplete_requirement "The specified - '${DB_BIN_DIR}' does
not contain the database server binaries.";
| fi
| if [ -z "${DB_LIB_DIR}" ]; then
DB_LIB_DIR=${DB_INSTALL_DIR}/lib; fi
| if [ -z "${DB_DESC}" ]; then DB_DESC=`${DB_BIN_DIR}/pg_config
--version`; fi
| if [ -z "${DB_DATA_DIR}" ]; then
| _die_incomplete_requirement "Missing data directory settings in the
script. Please set 'DB_DATA_DIR' variable in the script.";
| fi
| if [ ! -d "${DB_DATA_DIR}" ]; then
| _die_incomplete_requirement "The specified - '${DB_DATA_DIR}' is
not a valid. It is not present on the system.";
| fi
| if [ ! -f "${DB_DATA_DIR}/PG_VERSION" -o ! -d
"${DB_DATA_DIR}/base" -o ! -d "${DB_DATA_DIR}/global" ]; then
| _die_incomplete_requirement "The directory - '${DB_DATA_DIR}'
does not look like a valid PostgreSQL/Postgres Plus Advanced Server
data directory."
| fi
| if [ -z "${DB_SERVICE_USER}" ]; then
| _die_incomplete_requirement "The service-user is not specified in
the service script. Please set 'DB_SERVICE_USER' variable in the
script."
| fi
| DB_VALID_SERVICE_USER=`cat /etc/passwd | grep
"^${DB_SERVICE_USER}:"`
| if [ -z "${DB_VALID_SERVICE_USER}" ]; then
| _die_incomplete_requirement "The service-user
'${DB_SERVICE_USER}' is not present on the system. Please specify
the correct information."
| fi
| DB_DATA_DIR_OWNER=`ls -l ${DB_DATA_DIR}/PG_VERSION | awk
'{print $3}'`
| if [ x"${DB_DATA_DIR_OWNER}" != x"${DB_SERVICE_USER}" ]; then
| _die_incomplete_requirement "The specified user -
'${DB_SERVICE_USER}' does not own the data directory -
'${DB_DATA_DIR}'. The data directory is owned by the user -
'${DB_DATA_DIR_OWNER}'."
| fi
| if [ -z "${DB_HBA_FILE}" ]; then
DB_HBA_FILE=${DB_DATA_DIR}/pg_hba.conf; fi
| if [ ! -f "${DB_HBA_FILE}" ]; then
| _die_incomplete_requirement "The hba-file - '${DB_HBA_FILE}' does
not exist."
| fi
| if [ -z "${DB_CONF_FILE}" ]; then
DB_CONF_FILE=${DB_DATA_DIR}/postgresql.conf; fi
| if [ ! -f "${DB_CONF_FILE}" ]; then
| _die_incomplete_requirement "The config-file - '${DB_CONF_FILE}'
does not exist."
| fi
| if [ -z "${DB_PID_FILE}" ]; then
DB_PID_FILE=${DB_DATA_DIR}/postmaster.pid; fi
| if [ -z "${DB_STARTUP_LOG}" ]; then
DB_STARTUP_LOG=${DB_DATA_DIR}/pg_log/startup.log; fi
| DB_STARTUP_OPTIONS=""
| if [ x"${DB_CONF_FILE}" != x"${DB_DATA_DIR}/postgresql.conf" ];
then
| DB_STARTUP_OPTIONS="-c 'config_file=${DB_CONF_FILE}'"
| fi
| if [ x"${DB_HBA_FILE}" != x"${DB_DATA_DIR}/pg_hba.conf" ]; then
| DB_STARTUP_OPTIONS="${DB_STARTUP_OPTIONS} -c
'hba_file=${DB_HBA_FILE}'"

fi

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

| DB_STARTUP_OPTIONS="${DB_STARTUP_OPTIONS} -c
'external_pid_file=${DB_PID_FILE}'"
| fi
| if [ x"${DEBUG_VALIDATION}" = x"1" ]; then
| echo "Using these values in the scripts:"
| echo ""
| echo "DB_DESC : ${DB_DESC}"
| echo ""
| echo "DB_INSTALL_DIR : ${DB_INSTALL_DIR}"
| echo "DB_BIN_DIR : ${DB_BIN_DIR}"
| echo "DB_LIB_DIR : ${DB_LIB_DIR}"
| echo ""
| echo "DB_DATA_DIR : ${DB_DATA_DIR}"
| echo "DB_HBA_FILE : ${DB_HBA_FILE}"
| echo "DB_CONF_FILE : ${DB_CONF_FILE}"
| echo "DB_PID_FILE : ${DB_PID_FILE}"
| echo "DB_STARTUP_LOG : ${DB_STARTUP_LOG}"
| echo ""
| echo "DB_SERVICE_USER : ${DB_SERVICE_USER}"
| echo "DB_STARTUP_OPTIONS : ${DB_STARTUP_OPTIONS}"
| echo ""
| fi
| }
| DEBUG_VALIDATION=0
| # See how we were called.
| case "$1" in
| start)
| _validate_script
| start
| ;;
| stop)
| _validate_script
| stop
| ;;
| reload)
| _validate_script
| reload
| ;;
| restart)
| _validate_script
| restart
| ;;
| condrestart)
| _validate_script

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

2.8 Conclusion

EDB Postgres Enterprise Manager Installation Guide

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

- EDB designs, establishes coding best practices, reviews, and verifies input validation for the logon UI for EDB Postgres Enterprise Manager 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.

3.0 EDB Postgres Enterprise Manager Configuring pgBouncer for Use with PEM Agents

This document provides detailed information about using pgBouncer as a connection pooler for limiting the number of connections from the PEM Agent towards the Postgres Enterprise Manager (PEM) server on non-Windows machine:

- Preparing the PEM Database Server - Chapter 3 provides information about preparing the PEM database server to be used with pgBouncer.
- Configuring pgBouncer - Chapter 4 provides detailed information about configuring pgBouncer to make it work with the PEM database server.
- Configuring the PEM Agent – Chapter 5 provides detailed information about configuring a PEM agent to connect to pgBouncer.

For detailed information about using the PEM 7 web interface, please see the PEM Getting Started Guide.

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

3.1 The PEM Server - PEM Agent Connection Management Mechanism

Each PEM agent connects to the PEM database server using the SSL certificates for each individual user. For example, an agent with `ID#1` connects to the PEM database server using the `agent1` user.

Connecting to the PEM database without pgBouncer

Prior to PEM version 7.5, the following limitations did not allow use of the connection pooler between the PEM server and PEM agent:

- The PEM agent uses an SSL Certificate to connect the PEM database server.
- It uses an individual user identifier when connecting to the PEM database server.

EnterpriseDB has modified the PEM agent to allow the agent to use a common database user (instead of the dedicated agent users) to connect the PEM database server.

Connecting to pgBouncer.

We recommend using PgBouncer versions equal to or later than version 1.9.0 as the connection pooler. Since versions 1.9.0 or later support `cert` authentication; PEM Agents can connect to pgBouncer using SSL certificates.

3.2 Preparing the PEM Database Server

You must configure the PEM database server to make it work with PgBouncer; the following example demonstrates the steps required to configure the PEM database server.

1. Create a dedicated user named `pgbouncer` on the PEM database server. For example:

```
pem=# CREATE USER pgbouncer PASSWORD 'ANY_PASSWORD' LOGIN;
CREATE ROLE
```

2. Create a user named `pem_admin1` (a non-super user) with `pem_admin` and `pem_agent_pool role` membership on the PEM database server. For example:

   ```
   pem=# CREATE USER pem_admin1 PASSWORD 'ANY_PASSWORD' LOGIN
    CREATEROLE;
   CREATE ROLE
   pem=# GRANT pem_admin, pem_agent_pool TO pem_admin1;
   GRANT ROLE
   ```

3. Grant `CONNECT` privilege to the `pgbouncer` user on the `pem` database. For example:

   ```
   pem=# GRANT CONNECT ON DATABASE pem TO pgbouncer ;GRANT USAGE ON
    SCHEMA pem TO pgbouncer;
   GRANT
   ```

4. Grant `USAGE` privilege to the `pgbouncer` user for the `pem` schema on the pem database. For example:

   ```
   pem=# GRANT USAGE ON SCHEMA pem TO pgbouncer;
   GRANT
   ```

5. Grant `EXECUTE` privilege to the `pgbouncer` user on the `pem.get_agent_pool_auth(text)` function in the `pem` database. For example:

   ```
   pem=# GRANT EXECUTE ON FUNCTION pem.get_agent_pool_auth(text) TO
   pgbouncer;
   GRANT
   ```

6. Use the `pem.create_proxy_agent_user(varchar)` function to create a user named `pem_agent_user1` on the PEM database server. For example:

   ```
   pem=# SELECT pem.create_proxy_agent_user('pem_agent_user1');
   create_proxy_agent_user
   -------------------------
   (1 row)
   ```

   The function will create a user with the same name with a random password, and grant `pem_agent` and `pem_agent_pool` roles to the user. This allows pgBouncer to use a proxy user on behalf of the agent.

7. Add the following entries to the start of the `pg_hba.conf` file of the PEM database server; this will allow pgBouncer user to connect to the pem database using the md5 authentication method. For example:

   ```
   # Allow the PEM agent proxy user (used by
   # pgbouncer) to connect the to PEM server using
   # md5

   local pem pgbouncer,pem_admin1 md5
   ```

3.3 Configuring PgBouncer

You must configure PgBouncer to work with the PEM database server. In our example, we will run PgBouncer as the enterprisedb system user. The following steps outline the process of configuring pgBouncer (version >= 1.9).

1. Open a terminal window and navigate into the pgBouncer directory.
2. Change the owner of the etc directory for pgBouncer (where `pgbouncer.ini` resides) to `enterprisedb` , and change the directory permissions to `0700` . For example:

   ```
   $ chown enterprisedb:enterprisedb /etc/edb/pgbouncer1.9
   $ chmod 0700 /etc/edb/pgbouncer1.9
   ```

3. Change the contents of the `pgbouncer.ini` or `edb-pgbouncer.ini` file as follows:

```
[databases]
;; Change the pool_size according to maximum connections allowed
;; to the PEM database server as required.
;; 'auth_user' will be used for authenticate the db user (proxy
;; agent user in our case)

pem = port=5444 host=/tmp dbname=pem auth_user=pgbouncer
pool_size=80 pool_mode=transaction
* = port=5444 host=/tmp dbname=pem auth_user=pgbouncer
pool_size=10

[pgbouncer]
logfile = /var/log/edb/pgbouncer1.9/edb-pgbouncer-1.9.log
pidfile = /var/run/edb/pgbouncer1.9/edb-pgbouncer-1.9.pid
listen_addr = *
;; Agent needs to use this port to connect the pem database now
listen_port = 6432
;; Require to support for the SSL Certificate authentications
;; for PEM Agents
client_tls_sslmode = require
;; These are the root.crt, server.key, server.crt files present
;; in the present under the data directory of the PEM database
;; server, used by the PEM Agents for connections.
client_tls_ca_file = /var/lib/edb/as11/data/root.crt
client_tls_key_file = /var/lib/edb/as11/data/server.key
client_tls_cert_file = /var/lib/edb/as11/data/server.crt
;; Use hba file for client connections
auth_type = hba
;; Authentication file, Reference:
;; https://pgbouncer.github.io/config.html#auth_file
auth_file = /etc/edb/pgbouncer1.9/userlist.txt
;; HBA file
auth_hba_file = /etc/edb/pgbouncer1.9/hba_file
;; Use pem.get_agent_pool_auth(TEXT) function to authenticate
;; the db user (used as a proxy agent user).
auth_query = SELECT * FROM pem.get_agent_pool_auth($1)
;; DB User for administration of the pgbouncer
admin_users = pem_admin1
;; DB User for collecting the statistics of pgbouncer
stats_users = pem_admin1
server_reset_query = DISCARD ALL
;; Change based on the number of agents installed/required
max_client_conn = 500
;; Close server connection if its not been used in this time.
;; Allows to clean unnecessary connections from pool after peak.
server_idle_timeout = 60
```

4. Use the following command to create and update the `/etc/edb/pgbouncer1.9/userlist.txt` authentication file for PgBouncer.

```
pem=# COPY (
SELECT 'pgbouncer'::TEXT, 'pgbouncer_password'
UNION ALL
SELECT 'pem_admin1'::TEXT, 'pem_admin1_password'
TO '/etc/edb/pgbouncer1.9/userlist.txt'
WITH (FORMAT CSV, DELIMITER ' ', FORCE_QUOTE *);

COPY 2
```

NOTE: A super user cannot invoke the PEM authentication query function `pem.get_proxy_auth(text)`

. If the `pem_admin` user is a super user, you must add the password to the authentication file, which is enterprisedb in the above example.

5. Create an HBA file `(/etc/edb/pgbouncer1.9/hba_file)` for PgBouncer that contains the following content:

   ```
   # Use authentication method md5 for the local connections to
   # connect pem database & pgbouncer (virtual) database.
   local pgbouncer all md5
   # Use authentication method md5 for the remote connections to
   # connect to pgbouncer (virtual database) using enterprisedb
   # user.

   host pgbouncer,pem pem_admin1 0.0.0.0/0 md5

   # Use authentication method cert for the TCP/IP connections to
   # connect the pem database using pem_agent_user1

   hostssl pem pem_agent_user1 0.0.0.0/0 cert
   ```

6. Change the owner of the HBA file `(/etc/edb/pgbouncer1.9/hba_file)` to `enterprisedb` , and change the directory permissions to `0600` . For example:

   ```
   $ chown enterprisedb:enterprisedb /etc/edb/pgbouncer1.9/hba_file
   $ chmod 0600 /etc/edb/pgbouncer1.9/hba_file
   ```

7. Enable the PgBouncer service, and start the service. For example:

   ```
   $ systemctl enable edb-pgbouncer-1.9

   Created symlink from
   /etc/systemd/system/multi-user.target.wants/edb-pgbouncer-1.9.service
   to /usr/lib/systemd/system/edb-pgbouncer-1.9.service.

   $ systemctl start edb-pgbouncer-1.9
   ```

3.4 Configuring the PEM Agent

You can use an RPM package to install a PEM Agent; for detailed installation information, please see the *PEM Installation Guide*, available from the EntepriseDB website at:

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

Configuring a New PEM Agent (installed using an RPM)

After using an RPM package to install the PEM agent, you will need to configure it to work it against a particular PEM database server. Use the following command:

```
$ PGSSLMODE=require PEM_SERVER_PASSWORD=pem_admin1_password
/usr/edb/pem/agent/bin/pemworker --register-agent --pem-server
pem_agent_user1 --display-name "Agent Name"

Postgres Enterprise Manager Agent registered successfully!
```

In above command, the command line argument `--pem-agent-user` instructs the agent to create an SSL certificate and key pair for the `pem_agent_user1` database user in `/root/.pem` directory.

For example:

```
/root/.pem/pem_agent_user1.crt

/root/.pem/pem_agent_user1.key
```

They will be used by the PEM agent to connect to the PEM database server as `pem_agent_user1` . It will also create an agent configuration file named `/usr/edb/pem/agent/etc/agent.cfg.`

You will find a line mentioning the agent-user to be used in the `agent.cfg` configuration file.

For example:

```
$ cat /usr/edb/pem/agent/etc/agent.cfg
[PEM/agent]
pem_host=172.16.254.22
pem_port=6432
agent_id=12
agent_user=pem_agent_user1
agent_ssl_key=/root/.pem/pem_agent_user1.key
agent_ssl_crt=/root/.pem/pem_agent_user1.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
allow_server_restart=true
max_connections=0
connect_timeout=-1
connection_lifetime=0
allow_batch_probes=false
heartbeat_connection=false
```

Configuring an Existing PEM Agent (installed using an RPM)

If you are using an existing PEM agent, you can copy the SSL certificate and key files to the target machine, and reuse the files. You will need to modify the files, adding a new parameter and replacing some parameters in the existing `agent.cfg` file:

Add a line for agent_user to be used for the agent. For example:

```
agent_user=pem_agent_user1
```

Update the port to specify the pgBouncer port. For example:

```
pem_port=6432
```

Update the certificate and key path locations. For example:

```
agent_ssl_key=/root/.pem/pem_agent_user1.key
agent_ssl_crt=/root/.pem/pem_agent_user1.crt
```

Please note: as an alternative, you can run the agent self registration, but that will create a new agent id. If you do run the agent self-registration, you must replace the new agent id with existing id, and disable the entry for the new agent id in the `pem.agent` table.For example:

```
pem=# UPDATE pem.agent SET active = false WHERE id = <new_agent_id>;
```

```
UPDATE 1
```

NOTE: Keep a backup of the existing SSL certificate, key file, and agent configuration file.

3.5 Conclusion

PEM Configuring PgBouncer for Use with PEM Agents

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

- EDB designs, establishes coding best practices, reviews, and verifies input validation for the logon UI for EDB Postgres Enterprise Manager 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.

4.0 PEM Upgrade and Migration

This guide provides detailed information about upgrading the Postgres nterprise Manager (PEM) server:

- **Upgrading a PEM Installation** - This section 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** - 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.
- **Uninstalling PEM** –This section provides detailed information about uninstalling PEM components.

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

4.1 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.

4.2 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 <moving_pem_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:

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 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:

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 `vcredist_x86.exe`).

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

http://www.enterprisedb.com/downloads/component-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, 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` .

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>/`
- 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>/`
- 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, 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.

4.3 Moving the Postgres Enterprise ManagerServer

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:

> 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` .

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.

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` .

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
```

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
```

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.

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 .

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.

4.4 Troubleshooting

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.

4.5 Uninstalling Postgres Enterprise Manager

The process of uninstalling the PEM server or agent is platform-specific. The name of the package for PEM server is `edb-pem` and for PEM agent is `edb-pem-agent` .

If you uninstall the PEM server package from a host, the PEM agent package installed on the same host doesn't get uninstalled. But if you uninstall the PEM agent package, then the PEM server package installed on the same host also gets uninstalled.

Uninstalling PEM from Windows hosts

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.

You can also invoke the uninstaller that resides at the following location:

```
C:\\Program Files\edb\pem\server
```

Uninstalling PEM from CentOS or RHEL hosts

You can use variations of the `rpm` , `yum remove` , or `yum erase` commands to remove the installed packages. Note that removing a package does not damage the PEM data directory.

- Include the `-e` option when invoking the rpm command to remove an installed package; the command syntax is:

  ```
  rpm -e <package_name>
  ```

- You can use the `yum remove` command to remove the pem server or agent package installed by yum. To remove a package, open a terminal window, assume superuser privileges, and enter the command:

  ```
  yum remove <package_name>
  ```

- You can use the `yum erase` command to remove the pem server or agent package along with the `edb-pem` and `edb-pem-docs` dependencies. To remove a package, open a terminal window, assume superuser privileges, and enter the command:

  ```
  yum erase <package_name>
  ```

Where *package_name* is the name of the package that you would like to remove.

Uninstalling PEM from Debian or Ubuntu hosts

You can use `apt-get remove` or `apt-get purge` command to uninstall the PEM server or agent package from a Debian or Ubuntu host:

- To uninstall PEM server or agent from a Debian or Ubuntu host without impacting the configuration files and data directories, invoke the following command:

  ```
  apt-get remove <package_name>
  ```

- To uninstall PEM server or agent along with the configuration files and data directory, invoke the following command:

  ```
  apt-get purge <package_name>
  ```

Where *package_name* is the name of the package that you would like to remove.

Uninstalling PEM from SLES hosts

To uninstall PEM server or agent from a SLES host, invoke the follwoing command:

```
zypper remove <package_name>
```

Where *package_name* is the name of the package that you would like to remove.

4.6 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

- EDB designs, establishes coding best practices, reviews, and verifies input validation for the logon UI for EDB Postgres Enterprise Manager 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.

5.0 PEM Agent User Guide

PEM is composed of three primary components: PEM server, PEM agent, and PEM web interface. The PEM agent is responsible for performing tasks on each managed machine and collecting statistics for the database server and operating system.

For information about the platforms and versions supported by PEM, visit the EnterpriseDB website at:

> https://www.enterprisedb.com/services-support/edb-supported-products-and-platforms#pem

This document provides information that is required to work with PEM agents. The guide will acquaint you with the basic registering, configuration, and management of agents. The guide is broken up into the following core sections:

- **Postgres Enterprise Manager - Overview** - This section provides an overview of PEM architecure and also provides information about hardware and software prerequsites for installing a PEM agent.
- **Registering a PEM Agent** - This section provides information about registration of a PEM agent.
- **Managing a PEM agent** - This section provides information about configuring and managing a PEM agent.
- **Troubleshooting for PEM agent** - This section provides information about trobleshooting for PEM agents.
- **Uninstalling a PEM agent** - This section provides information about uninstalling a PEM agent.

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

5.1 Postgres Enterprise Manager - Overview

PEM Architecture

PEM architecture

PEM components

PEM overview

Postgres Enterprise Manager (PEM) consists of components that provide the management and analytical features of PEM:

- **PEM Server**: The PEM server is used as the data repository for monitoring data and as a server to which both agents and clients connect. The PEM server consists of an instance of PostgreSQL, an associated database for storage of monitoring data, and a server that provides web services.

- **PEM web interface**: The PEM web interface allows you to manage and monitor Postgres servers and utilize PEM extended functionality. The web interface software is installed with the PEM server installer, and is accessed via your choice of web browser.
- **PEM Agent**: The PEM agent is responsible for executing tasks and reporting statistics from the agent host and monitored Postgres instances to the PEM server. A single PEM agent can monitor multiple installed instances of Postgres that reside on one or many hosts.
- **SQL Profiler plugin**: This plugin to the Postgres server is used to generate the monitoring data used by the SQL Profiler tool. Installation of the SQL Profiler plugin is optional, but the plugin must be installed into each instance of Postgres you wish to profile. The SQL Profiler may be used with any supported version of an EnterpriseDB distribution of a PostgreSQL server or an Advanced Server (not just those managed through the PEM server).

The PEM Agent installer creates two executables: the PEM worker (`pemworker.exe`) and the PEM agent (`pemagent.exe`). Each PEM worker has a corresponding PEM agent that you can use to start or stop the PEM worker. The PEM agent will also restart the PEM worker should it terminate unexpectedly. The PEM worker log file contains information related to PEM worker activity (probe activities, heartbeat responses, etc.), and is stored in `/var/log/pem/worker.log` .

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

A typical PEM installation.

5.2 Installing a PEM Agent

You can use a graphical installer to install the Postgres Enterprise Manager agent on a Windows host. This graphical installer can also be invoked from command line.

To install the Postgres Enterprise Manager agent on a Linux host, you must use an RPM package.

Installers are available from the EnterpriseDB website at:

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

Installing an Agent on a Windows Host

Installing agent on Windows

On a Windows system, 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.

PEM Agent Installation - Welcome window

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

PEM Agent Installation - License Agreement

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

PEM Agent Installation - Installation Directory

By default, the PEM agent is installed in the `C:\Program Files (x86)\edb\pem` directory. You can accept the default installation directory, or modify the contents of the Installation Directory field, specifying an alternate installation directory for the PEM agent.

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

Check the box next to `Register now?` to instruct the installer to register the newly installed PEM agent with the PEM server. Click `Next` to continue to the `PEM Server Installation Details` dialog.

PEM Agent Installation - PEM server installation details

Enter the connection details for the PEM server on the PEM server installation details dialog:

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

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

Note

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

PEM Agent Installation - pemAgent Service Account password

Provide the password for the `edb` account under which the pemAgent service will run.The agent certificate and key files will be create in its `C:\Users\edb\AppData\Roaming\pem` directory. Click `Next` to continue to `Agent Details` dialog.

PEM Agent Installation - Agent Details

The tree control displayed in the Browser panel of the PEM web interface displays the value entered in the `Description` field to identify the PEM agent. Specify a descriptive name for the agent, such as the hostname of the machine the agent is installed on, or a name that reflects the host's functionality. Provide a descriptive name, or accept the default provided by the PEM agent host, and click `Next` to continue.

If you checked the `Show advanced options` checkbox, the `Advanced options` dialog opens:

PEM Agent Installation - Advanced Options - Certificate Path

By default, the PEM agent installer places the certificate in the `C:\Program Files (x86)\edb\pem` directory. Specify an alternate path for the certificate or accept the default and click `Next` . The wizard is now ready to install the PEM agent; click `Back` to amend the installation directory, or `Next` to continue.

PEM Agent Installation - Ready to Install

Click `Next` on the `Ready to Install` dialog to instruct the installer to copy files to the system and register the agent on the PEM server.

PEM Agent Installation - Installing Progress

The PEM agent installer displays progress bars to mark the PEM agent's installation progress.

PEM Agent Installation - Finish

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

Invoking a Graphical Installer from the Command Line

Invoking graphical installer from CLI

You can include the `--mode unattended` option when invoking the installer to perform an installation without additional user input.

For a complete reference guide to the command line options, include the `--help` option when you invoke the installer.

Invoking a Graphical Installer in Unattended Mode

Invoking graphical installer in unattended mode

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

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

- install the PEM server; the `pg_hba.conf` file of the PEM server must allow connections from the host of the PEM agent.
- ensure that the monitored Postgres database has SSL enabled, and is accepting connections.

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

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

Where: `x.x` specifies the version of PEM agent. `pem_server_host_address` specifies the IP address of the host of the PEM server. `pem_server_port` specifies the port used by the backing PEM database; by default, the database uses port 5432. `pguser_password` specifies the password associated with the PEM database superuser. `agent_name` specifies a descriptive name for the PEM agent.

Installing an Agent on a RHEL or CentOS host

Installing agent on RHEL

Installing agent on CentOS

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

- Install the `epel-release` package on the host:

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

  Note

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

- 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-3images/Repository%20Access%2004-09-2019.pdf

After receiving your repository credentials you can:

1. Create the repository configuration file.
2. Modify the file, providing your user name and password.
3. Install `edb-pem-agent` .

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
```

Installing PEM Agent

After saving your changes to the configuration file, you can use the yum install command to install `edb-pem-agent` . For example, the following command installs `edb-pem-agent` :

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

When the installation is complete, `yum` will display a list of the installed packages and dependencies.

Using an RPM package to install the PEM agent

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

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

Installing an Agent on a SLES Host

Installing agent on SLES

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

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

SLES packages are available from:

https://zypp.enterprisedb.com

Before installing a PEM agent, you must install prerequisite packages.

Use the following commands in the given sequence to install the agent:

```
SUSEConnect -p sle-module-legacy/12/x86_64

SUSEConnect -p sle-sdk/12/x86_64

zypper addrepo https://download.opensuse.org/repositories/Apache:Modules/<SLE_version_service_p

zypper addrepo http://download.opensuse.org/repositories/Cloud:/OpenStack:/Newton:/cisco-
apic:/2.3.1/<SLE_version_service_pack>/ pem_opensuse_boost

zypper refresh

zypper install edb-pem-agent
```

Where `SLE_version_service_pack` is the version and service pack of the SLES that you are using, such as SLE_12_SP2 or SLE_12_SP3.

Installing an Agent on a Debian or Ubuntu Host

Installing agent on Debian

Installing agent on Ubuntu

To install PEM agent on a Debian or Ubuntu host, you must have credentials that allow access to the EnterpriseDB repository. To request credentials for the repository, contact EnterpriseDB .

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

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

   ```
   sudo su –
   ```

2. Configure the EnterpriseDB repository:

   ```
   sh -c 'echo "deb https://<username>:<password>@apt.enterprisedb.com/$(lsb_release - cs)-e
   ```

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

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

4. Add the EBD signing key:

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

5 . Update the repository metadata:

```
apt-get update
```

6. Use the following command to install the Debian package for PEM agent:

   ```
   apt-get install edb-pem-agent
   ```

5.3 Registering an Agent

Each PEM agent must be *registered* with the PEM server. The registration process provides the PEM server with the information it needs to communicate with the agent. The PEM agent graphical installer for Windows supports self-registration for the agent. You must use the `pemworker` utility to register the agent if the agent is on a Linux host.

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

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

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

Option	Description
`--pem-server`	Specifies the IP address of the PEM backend database server. This parameter i
`--pem-port`	Specifies the port of the PEM backend database server. The default value is 543
`--pem-user`	Specifies the name of the Database user (having superuser privileges) of the PE
`--pem-agent-user`	Specifies the agent user to connect the PEM server backend database server.
`--cert-path`	Specifies the complete path to the directory in which certificates will be created.
`--config-dir`	Specifies the directory path where configuration file can be found. The default is
`--display-name`	Specifies a user-friendly name for the agent that will be displayed in the PEM Br
`--force-registration`	Include the force_registration clause to instruct the PEM server to register the ag
`--group`	The name of the group in which the agent will be displayed.
`--team`	The name of the database role, on the PEM backend database server, that shou
`--owner`	The name of the database user, on the PEM backend database server, who will
`--allow_server_restart`	Enable the allow-server_restart parameter to allow PEM to restart the monitored

Option	Description
`--allow-batch-probes`	Enable the allow-batch-probes parameter to allow PEM to run batch probes on t
`--batch-script-user`	Specifies the operating system user that should be used for executing the batch
`--enable-heartbeat-connection`	Enable the enable-heartbeat-connection parameter to create a dedicated heartb
`--enable-smtp`	Enable the enable-smtp parameter to allow the PEM agent to send the email on
`--enable-snmp`	Enable the enable-snmp parameter to allow the PEM agent to send the SNMP t
`-o`	Specify if you want to override the configuration file options.

If you want to use any PEM feature for which database server restart is required by the pemagent such as Audit Manager, Log Manager, or Tuning Wizard, then you must set the value for `allow_server_restart` as `true` in the `agent.cfg` file.

Note

When configuring a shell/batch script run by a PEM agent that has PEM 7.11 or later version installed, the user for the `batch_script_user` parameter must be specified. 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` .

You can use the PEM_SERVER_PASSWORD environment variable to set the password of the PEM Admin User. If the PEM_SERVER_PASSWORD is not set, the server will use the PGPASSWORD or pgpass file when connecting to the PEM Database Server.

Failure to provide the password will result in a password authentication error; you will be prompted for any other required but omitted information. When the registration is complete, the server will confirm that the agent has been successfully registered.

Setting PEM Agent Configuration Parameters

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

```
heartbeat_connection = true
```

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

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

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

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

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

```
systemctl start pemagent
```

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

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

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

Using a non-root User Account to Register a PEM Agent

To register a PEM agent using a non-root user, you first need to install PEM agent as a root user. After installation, assume the identity of a non-root user (for example edb) and perform the following steps:

1. Create the `.pem` directory and `logs` directory as following and assign read, write, and execute permissions to the file:

```
mkdir /home/<edb>/.pem
mkdir /home/<edb>/.pem/logs
chmod 700 /home/<edb>/.pem
chmod 700 /home/<edb>/.pem/logs
```

2. Register the agent with PEM server using the `pemworker` utility as following:

```
./pemworker --register-agent --pem-server <172.19.11.230> --pem-user <postgres> --pem-port <5432> --display-name <non_root> --cert-path /home/<edb> --config-dir /home/<edb>
```

The above command creates agent certificates and an agent configuration file (`agent.cfg`) in the `/home/edb/.pem` directory. Assign read and write permissions to these files using the command:

```
chmod -R 600 /home/edb/.pem/agent*
```

3. Change the parameters of the `agent.cfg` file as following:

```
agent_ssl_key=/home/edb/.pem/agent<id>.key
agent_ssl_crt=/home/edb/.pem/agent<id>.crt
log_location=/home/edb/.pem/worker.log
agent_log_location=/home/edb/.pem/agent.log
```

4. Update the value for path and user in the `pemagent` service file:

- If you are using RHEL or CentOS 6, update the pemagent service file to reflect the correct path of `agent.cfg` file and also change user `su` to `su edb` .
- If you are using RHEL or CentOS 7, update the parameters as following:

```
User=edb
ExecStart=/usr/edb/pem/agent/bin/pemagent -c /home/edb/.pem/agent.cfg
```

5. Kill the agent process that was started earlier, and then restart the agent service using the non-root user as follows:

- If you are using RHEL or CentOS 6, `sudo /etc/init.d/pemagent start/stop/restart`
- If you are using RHEL or CentOS 7, `sudo systemctl start/stop/restart pemagent`

6. Check the agent status on PEM dashboard.

5.4 Managing a PEM Agent

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

Agent Privileges

Agent Privileges

By default, the PEM agent is installed with `root` privileges for the operating system host and superuser privileges for the database server. These privileges allow the PEM agent to invoke unrestricted probes on the monitored host and database server about system usage, retrieving and returning the information to the PEM server.

Please note that PEM functionality diminishes as the privileges of the PEM agent decrease. For complete functionality, the PEM agent should run as `root` . If the PEM agent is run under the database server's service account, PEM probes will not have complete access to the statistical information used to generate reports, and functionality will be limited to the capabilities of that account. If the PEM agent is run under another lesser-privileged account, functionality will be limited even further.

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

Note

The above-mentioned list is not comprehensive, but should provide an overview of the type of functionality that will be limited.

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

Probe	Operating System	PEM Functionality Affected
Audit Log Collection	Linux/Windows	PEM will receive empty data from the PEM database.
Server Log Collection	Linux/Windows	PEM will be unable to collect server log information.
Database Statistics	Linux/Windows	The Database/Server Analysis dashboards will contain incomplete info
Session Waits/System Waits	Linux/Windows	The Session/System Waits dashboards will contain incomplete informa
Locks Information	Linux/Windows	The Database/Server Analysis dashboards will contain incomplete info
Streaming Replication	Linux/Windows	The Streaming Replication dashboard will not display information.
Slony Replication	Linux/Windows	Slony-related charts on the Database Analysis dashboard will not disp
Tablespace Size	Linux/Windows	The Server Analysis dashboard will not display complete information.
xDB Replication	Linux/Windows	PEM will be unable to send xDB alerts and traps.

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

If the probe is querying the database with insufficient privileges, the probe may return a `permission denied` error or display the returned data in a PEM chart or graph as an empty value.

When a probe fails, an entry will be written to the log file that contains the name of the probe, the reason the probe failed, and a hint that will help you resolve the problem.

You can view probe-related errors that occurred on the server in the Probe Log Dashboard, or review error messages in the PEM worker log files. On Linux, the default location of the log file is:

```
/var/log/pem/worker.log
```

On Windows, log information is available on the `Event Viewer` .

Agent Configuration

Agent Configuration

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

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

On Linux systems, PEM configuration options are stored in the `agent.cfg` file, located in `/usr/edb/pem/agent/etc` . The `agent.cfg` file contains the following entries:

Parameter Name	Description
pem_host	The IP address or hostname of the PEM server.
pem_port	The database server port to which the agent connects to communicate with the PEM server.
pem_agent	A unique identifier assigned to the PEM agent.
agent_ssl_key	The complete path to the PEM agent's key file.
agent_ssl_crt	The complete path to the PEM agent's certificate file.
agent_flag_dir	Used for HA support. Specifies the directory path checked for requests to take over monitorin
log_level	Log level specifies the type of event that will be written to the PEM log files.

Parameter Name	Description
log_location	Specifies the location of the PEM worker log file.
agent_log_location	Specifies the location of the PEM agent log file.
long_wait	The maximum length of time (in seconds) that the PEM agent will wait before attempting to co
short_wait	The minimum length of time (in seconds) that the PEM agent will wait before checking which p
alert_threads	The number of alert threads to be spawned by the agent.
enable_smtp	When set to true, the SMTP email feature is enabled.
enable_snmp	When set to true, the SNMP trap feature is enabled.
enable_nagios	When set to true, Nagios alerting is enabled.
connect_timeout	The max time in seconds (a decimal integer string) that the agent will wait for a connection.
allow_server_restart	If set to TRUE, the agent can restart the database server that it monitors. Some PEM features
max_connections	The maximum number of probe connections used by the connection throttler.
connection_lifetime	Use ConnectionLifetime (or connection_lifetime) to specify the minimum number of seconds a
allow_batch_probes	If set to TRUE, the user will be able to create batch probes using the custom probes feature.
heartbeat_connection	When set to TRUE, a dedicated connection is used for sending the heartbeats.
batch_script_dir	Provide the path where script file (for alerting) will be stored.
connection_custom_setup	Use to provide SQL code that will be invoked when a new connection with a monitored server
ca_file	Provide the path where the CA certificate resides.
batch_script_user	Provide the name of the user that should be used for executing the batch/shell scripts.

On 64 bit Windows systems, PEM registry entries are located in:

```
HKEY_LOCAL_MACHINE\Software\Wow6432Node\EnterpriseDB\PEM\agent
```

The registry contains the following entries:

Agent Properties

Agent Properties

The `PEM Agent Properties` dialog provides information about the PEM agent from which the dialog was opened; to open the dialog, right-click on an agent name in the PEM client tree control, and select `Properties` from the context menu.

The PEM Agent Properties dialog

Use fields on the PEM Agent properties dialog to review or modify information about the PEM agent:

- The `Description` field displays a modifiable description of the PEM agent. This description is displayed in the tree control of the PEM client.
- You can use groups to organize your servers and agents in the PEM client tree control. Use the `Group` drop-down listbox to select the group in which the agent will be displayed.
- Use the `Team` field to specify the name of the group role that should be able to access servers monitored by the agent; the servers monitored by this agent will be displayed in the PEM client tree control to connected team members. Please note that this is a convenience feature. The Team field does not provide true isolation, and should not be used for security purposes.
- The `Heartbeat interval` fields display the length of time that will elapse between reports from the PEM agent to the PEM server. Use the selectors next to the `Minutes` or `Seconds` fields to modify the interval.

5.5 PEM Agent Troubleshooting

Restoring a Deleted PEM Agent

restoring a deleted agent

If an agent has been deleted from the `pem.agent` table then you cannot restore it. You will need to use the pemworker utility to re-register the agent.

If an agent has been deleted from PEM Web client but still has an entry in the `pem.agent` table with value of active = f, then you can restore the agent using the following steps:

1. Use the following command to check the values of the `id` and `active` fields:

   ```
   pem=# SELECT * FROM pem.agent;
   ```

2. Update the status for the agent to `true` in the `pem.agent` table:

   ```
   pem=# UPDATE pem.agent SET active=true WHERE id=<x>;
   ```

   Where, `x` is the identifier that was displayed in the output of the query used in step 1.

3. Refresh the PEM web client.

The deleted agent will be restored again. However, the servers that were bound to that particular agent might appear to be down. To resolve this issue, you need to modify the PEM agent properties of the server to add the bound agent again; after the successful modification, the servers will be displayed as running properly.

Reconfiguring the PEM Server

reconfiguring an agent

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'
   ```

Using the Command Line to Delete a PEM Agent with Down or Unknown Status

deleting agents with Down status

deleting agents with Unknown status

Using the PEM web interface to delete PEM agents with `Down` or `Unknown` status may be difficult if the number of such agents is large. In such situations, you might want to use the command line interface to delete Down or Unknown agents.

1. Use the following query to delete the agents that are `Down` for more than *N* number of hours:

```
UPDATE pem.agent SET active=false WHERE id IN
(SELECT a.id FROM pem.agent
a JOIN pem.agent_heartbeat b ON (b.agent_id=a.id)
WHERE a.id IN
(SELECT agent_id FROM pem.agent_heartbeat WHERE (EXTRACT (HOUR FROM now())-
EXTRACT (HOUR FROM last_heartbeat)) > <N> ));
```

2. Use the following query to delete the agents with an `Unknown` status:

```
UPDATE pem.agent SET active=false WHERE id IN
(SELECT id FROM pem.agent WHERE id NOT IN
(SELECT agent_id FROM pem.agent_heartbeat));
```

5.6 'Uninstalling a PEM Agent'

Use the uninstaller provided in the PEM installation directory to remove PEM agent from a system. By default, the PEM agent uninstaller is located:

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

```
./uninstall-<agent_name>
```

Where *agent_name* is the name of the agent that you wish to remove.

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

5.7 Conclusion

EDB Postgres Enterprise Manager Agent User Guide

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

- EDB designs, establishes coding best practices, reviews, and verifies input validation for the logon UI for EDB Postgres Enterprise Manager 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.

6.0 PEM BART Management Guide

This guide will acquaint you with the dialogs that are built into the Postgres Enterprise Manager (PEM) web interface that make it easier for you to monitor and manage BART.

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

6.1 Managing a BART Server

Managing a BART Server

Postgres Enterprise Manager (PEM) is designed to assist database administrators, system architects, and performance analysts when administering, monitoring, and tuning PostgreSQL and Advanced Server database servers.

The EDB Backup and Recovery Tool (BART) is an administrative utility providing simplified backup and recovery management for multiple local or remote EDB Postgres Advanced Server and PostgreSQL database servers. For more information about BART, please visit the EnterpriseDB website at:

> https://www.enterprisedb.com/enterprise-postgres/edb-postgres-backup-and-recovery-tool

From PEM version 7.10 onwards, you can manage a BART server through PEM console. PEM provides a user-friendly interface that allows you to manage your BART server and perform all the BART operations from PEM console.

Prerequisites

Prerequisites

- Before adding a BART server to the PEM console, you must manually install and configure BART on the BART host. For more information about installing and configuring BART, please see the *BART Installation Guide* available at:

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

- Before associating a database server with a BART server, you must install SSH on the database server and the BART server.
- Before restoring a BART backup, you must install BART, PEM agent, and SSH on the target server. SSH must also be installed on the BART server that you plan to use for restore.
- To take a backup of the standby database servers, you must ensure that the latest *pg_basebackup* utility is installed on the database server that you want to manage through BART.

Configuring a BART Server

Configuring a BART Server

You can use the `Create–BART server` dialog to register an existing BART server with the PEM server. To access the dialog, right-click on the `BART Servers` node and select `Create-BART Server` .

The Create-BART server dialog - General tab

Use the fields on the `General` tab to describe the general properties of the BART Server:

- Use the `Agent Name` field to select the agent that you want to configure as a BART server. Only those PEM agents that are supported for BART are listed in the drop-down list.
- Use the `Server Name` field to specify a user-friendly name for the server. The name specified will identify the server in the Browser tree.
- Use the `Host` field to specify the IP address of the host or agent where BART is installed.
- Use the `User` field to specify the user name that will be used for performing all the BART operations. You can either use the `enterprisedb` (for Advanced Server) or `postgres` (for PostgreSQL) database user account or you can create a new BART user account. This user must be an operating system user who owns the BART backup catalog directory.
- Use the `Installation path` field to specify the directory path where BART is installed on the host or BART server.
- Use the `Backup path` field to specify the file system parent directory where all BART backups and archived WAL files will be stored.
- Use the `pg_basebackup_path` field to specify the path to the `pg_basebackup` utility.

- Use the `Xlog/WAL` method field to specify how the transaction log should be collected during the execution of pg_basebackup. The default option is `fetch` ; it specifies that the transaction log files will be collected after the backup has completed. Set the `Xlog` method to `stream` to stream the transaction log in parallel with the full base backup creation. If streaming is used, the `max_wal_senders` configuration parameter in the `postgresql.conf` file for affected database servers must account for an additional session for the streaming of the transaction log (the setting must be a minimum of 2).

  For more information about Xlog method, see:

  > https://www.postgresql.org/docs/current/app-pgbasebackup.html

- Use the `Retention policy` field to specify the retention policy for the backup. This determines when an active backup should be marked as obsolete, and hence, be a candidate for deletion. You can specify the retention policy in terms of number of backup or in terms of duration (days, weeks, or months).
- Use the `Log file` field to specify the path to BART log file. This is an optional field.

The Create-BART server dialog - Misc tab

Use the fields on the `Misc` tab to describe the backup-related properties of the BART Server:

- Use the `Scanner log file` field to specify the path to the Xlog/WAL scanner log file. This is an optional field; BART does not create a WAL scanner log file if you do not specify the path.
- Use the `WAL compression?` switch to specify if you want to compress the archived Xlog/WAL files in Gzip format. To enable WAL compression, the gzip compression program must be present in the BART user account's PATH. The WAL compression setting must not be enabled for those database servers where you need to take incremental backups.
- Use the `Copy WALs during restore?` field to specify how the archived WAL files are collected when invoking the RESTORE operation. Set to enabled to copy the archived WAL files from the BART backup catalog to the `restore_path/archived_wals` directory prior to the database server archive recovery. Set to `disabled` to retrieve the archived WAL files directly from the BART backup catalog during the database server archive recovery. Enabling this option helps you save time during the restore operation.
- Use the `Thread count` field to specify the number of worker threads for copying blocks or data files from the database server to the BART backup catalog. Specify a `thread count` of `1` if you want to take the backup using the `pg_basebackup` utility.
- Use the `Batch size` field to specify the number of blocks of memory used for copying modified blocks. This is applicable only for incremental backups.
- Use the `scan interval` field to specify the number of seconds after which the WAL scanner should scan the new WAL files.
- Use the `MBM scan timeout` field to specify the number of seconds to wait for MBM files before timing out. This is applicable only for incremental backups.

Associating the BART Server with a Database Server

Associating the BART Server with a Database Server

After configuring the BART server, you need to associate it with the database server whose backup you want to manage with BART. You can do one of the following:

- Use the PEM console to modify the properties of an existing monitored database server to map it to the newly configured BART server.
- Use the PEM console to create a new monitored database server, and map it to the newly configured BART server.

To map the BART server to a new PEM database server, right-click the `PEM Server Directory` node and select `Create` > `Server` . Enter the details on all the generic tabs and then enter the BART-specific details on the `BART` tab.

The Create Server dialog (BART - General tab)

Use the fields on the `General` tab to describe the general properties of the BART Server that will map to the PEM server:

- Use the `BART server` field to select the BART server name. All the BART servers configured in the PEM console will be listed in this drop down list.
- Use the `Server name` field to specify a name for the database server that you want to backup using the BART server. This name gets stored in the BART configuration file.
- Use the `Description` field to specify the description of the database server.
- Use the `Backup name` field to specify a template for user-defined names to be assigned to the backups of the database server. If you do not specify a backup name template, then the backup can only be referenced in BART sub-commands by the BART assigned, integer backup identifier.
- Use the `Host address` field to specify the IP address of the database server that you want to configure for backup.
- Use the `Port` field to specify the port to be used for the database that you want to backup.
- Use the `User` field to specify the user of the database that you want to backup using BART through PEM console. If you want to enable incremental backups for this database server, then the user must be a superuser.
- Use the `Password` field to specify the password for the user of the database that you want to backup.
- Use the `Cluster Owner` field to specify the Linux operating system user account that owns the database cluster. This is typically `enterprisedb` for Advanced Server database clusters installed in the Oracle databases compatible mode, or `postgres` for PostgreSQL database clusters and for Advanced Server database clusters installed in the PostgreSQL databases compatible mode.
- Use the `Archive command` field to specify the desired format of the archive command string to be used in the `bart.cfg` file. Inputs provided for the Archive command will overwrite the database server's `Postgresql.conf` file. Once the server gets added, the database server will be restarted or database configurations will be reloaded.
- Use the `Allow incremental backup?` switch to specify if incremental backup should be enabled for this database server.
- Use the `Setup passwordless SSH?` switch to specify if you want to create SSH certificates to allow passwordless logins between the Database Server and the BART server. Ensure to bind a PEM agent before setting up the passwordless SSH authentication. Passwordless SSH will not work for a database server being remotely monitored by a PEM agent.

The Create - Server dialog (BART - Misc tab)

Use the fields on the `Misc` tab to describe the miscellaneous properties of the BART Server:

- Use the `Override default configuration?` Switch to specify if you want to override the BART server configurations with the specific database server configurations.
- Use the `Xlog` method to specify how the transaction log should be collected during the execution of `pg_basebackup` .
- Use the `Retention policy` field to specify the retention policy for the backup. This determines when an active backup should be marked as obsolete, and hence, be a candidate for deletion. You can specify the retention policy in terms of number of backup or in terms of duration (days, weeks, or months).
- Use the `WAL compression` switch to specify if you want to compress the archived Xlog/WAL files in Gzip format. To enable WAL compression, the gzip compression program must be present in the BART user account's PATH. The wal_compression setting must not be enabled for those database servers where you need to take incremental backups.
- Use the `Copy WALs during restore` field to specify how the archived WAL files are collected when invoking the RESTORE operation. Set to enabled to copy the archived WAL files from the BART backup catalog to the <restore_path>/archived_wals directory prior to the database server archive recovery. Set to disabled to retrieve the archived WAL files directly from the BART backup catalog during the database server archive recovery.
- Use the `Thread count` field to specify the number of threads to copy the blocks. You must set `thread count` to `1` if you want to take a backup with the `pg_basebackup` utility.
- Use the `Batch size` field to specify the number of blocks of memory used for copying modified blocks,

applicable only for incremental backups.

- Use the `Scan interval` field to specify the number of seconds after which the WAL scanner should scan the new WAL files.
- Use the `MBM scan timeout` field to specify the number of seconds to wait for MBM files before timing out, applicable only for incremental backups.

Viewing the BART Server Details on a PEM Dashboard

Viewing the BART Server Details on a PEM Dashboard

Once the BART server is associated with the database server, you can see the entire backup and restore related details for that particular BART server on the PEM Dashboard. You can also perform operations such as restoration or deletion of a backup that is listed on the dashboard.

The BART Backup Dashboard

When you select a monitored BART server, details of all the associated database servers along with their backups and restore activities are displayed as a chart on the Dashboard in the `Backup and Restore Activities` panel. You can filter the list of backups on any criteria that you specify in the filter boxes (the database server, activity, or duration).

The `Managed Database servers` panel displays a list of all the database servers managed by that particular BART server along with their high-level details.

The `Initiated Server Backups` panel displayes a list of all the backups of the database servers managed by that particular BART server. You can filter the list to display the details of a particular database server. You can also filter the list on any criteria that you specify in the filter box. Typically, this filter works with any kind of string value (excluding date, time, and size) listed under the columns. For example, you can type `tar` to filter the list and display only those backups that are in tar format.

Backup details displayed include the `Backup Name` , `Backup ID` , `Status` , `Server Name` , `Start Time` , `Type` , `Parent ID` , `Format` , `Duration` , and `Size` . The `Status` column shows the status of the backups which can be one of the following: `In Progress` , `Active` , `Keep` , or `Obsolete` .

The backups are marked as `Obsolete` after the backup retention period has passed or number of retained backups that you have specified as retention policy of the BART server is met. If you want to make an exception so that a particular backup does not get marked as `Obsolete` even after the expiry of the duration of retention policy, then you need to mark that particular backup as `Keep` . Similarly, if you mark a particular backup as `NoKeep` , the backup is re-evaluated to determine if its status should be changed back to obsolete based upon the current retention policy.

A pin in the first column under *Actions* indicates that a backup can be marked as *Keep* by clicking the pin; while an inverted pin indicates that the backup can be marked as *NoKeep*. The second column under `Actions` displays the `Restore` icon; you can perform the `Restore` operation by clicking on the icon.

You can delete all the `Obsolete` backups by clicking the `Delete Obsolete` button. You can also refresh the list of backups by clicking the `Refresh` button.

Scheduling BART Backups

Scheduling BART Backups

To schedule a backup using BART, select `Schedule Backup` under `Tools` menu. You can see a list of scheduled backups with details such as `Logs` , `Last result` , `Database server` , `Last backup name` , `Started on` , `Type` , `Parent` , `Format` , `Verify checksum?` , and `Use pg_basebackup?` . Click the Add icon (+) to add a schedule for the backup. Enter the details in the schedule definition dialog:

The Schedule Backup dialog - General options

Use the fields on the `General` tab to describe the general properties of the backup:

- Use the `Database Server` field to specify the target database server that you want to back up.
- Use the `Backup name` to specify a user-defined name for the backup.
- Use the `Backup type` switch to specify the backup type I. e. full backup or incremental backup.
- Use the `Parent backup` field to select the ID of the parent backup for incremental backup. This parent backup can either be a full or an incremental backup.
- Use the `Format switch` to specify the output format of the backup i.e plain text or tar. For incremental backup, you need to select plain text only.
- Use the `Gzip compression` switch to specify if gzip compression should be enabled for the backup. This option is applicable only for the tar format.
- Use the `Compression level` field to specify the gzip compression level on the tar file output.
- Use the `Thread count` field to specify the number of threads that will copy the blocks.
- Use the `MBM scan timeout` field to specify the number of seconds to wait for required MBM files before timing out.
- Use the `Verify checksum` field to specify if you want the application to verify the checksum of the backup.
- Use the `pg_basebackup` field to specify if the pg_basebackup utility should be used for the backup. Typically, pg_basebackup utility is used only for backing up the standby servers since it cannot be used for incremental backups.

The Schedule Backup dialog - (Schedule : General)

Provide information on the `Schedule` tab to describe the scheduling details:

- Use the `Enabled?` switch to indicate if the schedule should be enabled (`Yes`) or disabled (`No`).
- Use the calendar selector in the `Start` field to specify the starting date and time for the schedule.
- Use the calendar selector in the `End` field to specify the ending date and time for the schedule.

The Schedule Backup dialog - (Schedule : Repeat)

Use the fields on the `Repeat` tab to specify the details about the schedule in a cron-style format. The schedule will execute on each date or time element selected on the `Repeat` tab. Click within a field to open a list of valid values for that field; click on a specific value to add that value to the list of selected values for the field. To clear the values from a field, click the `X` located at the right-side of the field.

Use the fields within the `Days` box to specify the days on which the schedule will execute:

- Use the `Week Days` field to select the days on which the schedule will execute.
- Use the `Month Days` field to select the numeric days on which the schedule will execute. Specify the Last Day to indicate that the schedule should be performed on the last day of the month, regardless of the date.
- Use the `Months` field to select the months in which the schedule will execute.

Use the fields within the `Times` box to specify the times at which the schedule will execute:

- Use the `Hours` field to select the hour at which the schedule will execute.
- Use the `Minutes` field to select the minute at which the schedule will execute.

The Schedule Backup dialog - (Schedule : Notifications)

Use the fields on the `Notifications` tab to specify the email notification settings for a scheduled backup:

- Use the `Send the notifications` field to specify when you want the email notifications to be sent.
- Use the `Email group` field to specify the email group that should receive the email notification.

Restoring BART Backups

Restoring BART Backups

You can restore the backups that you have earlier created using BART server on a target remote host. When you select a particular BART server, all the associated backups are listed in the Dashboard under `Initiated Server Backups` .

To restore a backup, click the `Restore` icon next to the backup that you want to restore.

The Restore Backup dialog - General

In the `Restore Backup` dialog, provide information in the fields on the `General` tab:

- Use the `Target agent` field name to specify the name of the agent where you want to restore the backup.
- Use the `Remote user` field to specify the use account on the remote database server host where you want to restore the backup.
- Use the `Remote host address` field to specify the IP address of the remote host where you want to restore the backup.
- Use the `SSH port` field to specify the SSH port to be used for restoring the backup.
- Use the `Restore path` field to specify the path where you want to restore the backup.
- Use the `Number of workers` field to specify processes to run in parallel to stream the modified blocks of an incremental backup to the restore location.
- Use the `Setup passwordless SSH?` switch to specify if you want to create SSH certificates to allow passwordless logins between the BART server and the target host for restore.

The Restore Backup dialog - Advanced

On the `Advanced` tab, specify your preferences for advanced options for restoring the backup:

- Use the `Copy WALs to restore path?` switch to specify if you want to copy WALs to the restore path.
- Use the `Point in time recovery` switch to specify if you want point in time recovery.
- Use the `Timeline ID` field to specify the timeline ID to be used for replaying the archived WAL files for point-in-time recovery.
- Use the `Transaction ID (XID)` field to specify the transaction ID for point-in-time recovery.
- Use the `Timestamp` field to the timestamp to be used for restore.

Note

You can specify either `Transaction ID` or `Timestamp` for the point-in-time recovery.

The Restore Backup dialog - Notifications

Use the fields on the `Notifications` tab to specify the email notification settings for restoring the backup.

- Use the `Send the notifications` field to specify when you want the email notifications to be sent.
- Use the `Email group` field to specify the email group that should receive the email notification.

6.2 Conclusion

EDB Postgres Enterprise Manager BART Management Features Guide

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

- EDB designs, establishes coding best practices, reviews, and verifies input validation for the logon UI for EDB Postgres Enterprise Manager 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.

7.0 PEM Enterprise Features Guide

This guide will acquaint you with the tools and wizards that are built into the Postgres Enterprise Manager (PEM) web interface that make it easier for you to monitor and manage your system.

This guide is not a comprehensive resource; rather, it is meant to serve as an aid to help you evaluate the tool and bring you up to speed with the basics of how to use the product. For more detailed information about using PEM's functionality, please see the online help made available by the PEM client.

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

7.1 What's New

The following features have been added to create Postgres Enterprise Manager 7.13:

- Core Usage Report: The Core Usage report provides metrics such as total number of cores used by the EDB Postgres Advanced Server or Postgres database that is being managed by PEM. This report can help you understand if you are in compliance with core-based licensing guidelines.
- SMMP v3 support: SNMP v3 support enables the PEM Server and PEM Agent to use SNMP Version 3 for secure communication by authenticating and encrypting data packets over the network.
- Schema Diff tool (Beta): The Schema Diff tool allows you to compare two database schema objects and synchronize the two schemas. The tool lists the differences and also generates the synchronization scripts for the two schemas.
- Auto logout inactive users: An inactive user session is automatically logged out after a certain period of inactivity. This timeout interval can be configured in the `config.py` file. This feature improves security by reducing the unintended access to the database server.
- Accessibility Improvements: Enables screen-readers to read labels and descriptions of the non-text elements, to identify the alert errors, and to read relationship attributes in nested elements.
- Other features include:
  - Enhancements to the System Configuration Report.
  - Support for a multi-level partitioned table.

7.2 The PEM Query Tool

PEM Query Tool

PEM contains a feature-rich Interactive Development Environment (IDE) that allows you to issue ad-hoc SQL queries against Postgres servers.

You can access the Query Tool via the *Query Tool* menu option on the *Tools* menu, or through the context menu of select nodes of the Browser tree control. The Query Tool allows you to:

- Issue ad-hoc SQL queries.
- Execute arbitrary SQL commands.
- Edit the result set of a SELECT query if it is updatable.
- Displays current connection and transaction status as configured by the user.
- Save the data displayed in the output panel to a CSV file.
- Review the execution plan of a SQL statement in either a text, a graphical format or a table format (similar to https://explain.depesz.com).
- View analytical information about a SQL statement.

The PEM Query Tool

You can open multiple copies of the Query tool in individual tabs simultaneously. To close a copy of the Query tool, click the *X* in the upper-right hand corner of the tab bar.

The Query Tool features two panels:

- The upper panel displays the *SQL Editor*. You can use the panel to enter, edit, or execute a query. It also shows the *History* tab which can be used to view the queries that have been executed in the session, and a *Scratch Pad* which can be used to hold text snippets during editing. If the Scratch Pad is closed, it can be re-opened (or additional ones opened) by right-clicking in the SQL Editor and other panels and adding a new panel.
- The lower panel displays the *Data Output* panel. The tabbed panel displays the result set returned by a query, information about a query's execution plan, server messages related to the query's execution and any asynchronous notifications received from the server.

The Query Tool Toolbar

The *Query Tool* toolbar uses context-sensitive icons that provide shortcuts to frequently performed tasks. If an icon is highlighted, the option is enabled; if the icon is grayed-out, the task is disabled.

The Query Tool Toolbar

Hover over an icon to display a tool-tip that describes the icon's functionality:

The SQL Editor Panel

The *SQL editor* panel is a workspace where you can manually provide a query, copy a query from another source, or read a query from a file. The SQL editor features syntax coloring and auto-completion.

The SQL Editor Panel

To use auto-complete, begin typing your query; when you would like the Query editor to suggest object names or commands that might be next in your query, press the Control+Space key combination. For example, type "*SELECT * FROM* " (without quotes, but with a trailing space), and then press the Control+Space key combination to select from a popup menu of auto-complete options.

Using Auto-complete

After entering a query, select the *Execute/Refresh* icon from the toolbar. The complete contents of the SQL editor panel will be sent to the database server for execution. To execute only a section of the code that is displayed in the SQL editor, highlight the text that you want the server to execute, and click the *Execute/Refresh* icon.

Executing a command

The message returned by the server when a command executes is displayed on the *Messages* tab. If the command is successful, the *Messages* tab displays execution details.

The Messages Tab

Options on the *Edit* menu offer functionality that helps with code formatting and commenting:

- The auto-indent feature will automatically indent text to the same depth as the previous line when you press the Return key.
- Block indent text by selecting two or more lines and pressing the Tab key.
- Implement or remove SQL style or toggle C style comment notation within your code.

You can also **drag and drop** certain objects from the tree-view which can save time in typing long object names. Text containing the object name will be fully qualified with schema. Double quotes will be added if required. For functions and procedures, the function name along with parameter names will be pasted in the Query Tool.

The Data Output Panel

The *Data Output* panel displays data and statistics generated by the most recently executed query.

The Data Output Panel

Data Output Tab

The *Data Output* tab displays the result set of the query in a table format. You can:

- Select and copy from the displayed result set.
- Use the *Execute/Refresh* options to retrieve query execution information and set query execution options.
- Use the *Download as CSV* icon to download the content of the *Data Output* tab as a comma-delimited file.
- Edit the data in the result set of a SELECT query if it is updatable.

A result set is updatable if:

- All columns are either selected directly from a single table, or they are not actually a table column (e.g. concatenation of 2 columns). Only columns that are selected directly from the table are editable, other columns are read-only.
- All the primary key columns or OIDs of the table are selected in the result set.

Any columns that are renamed or selected more than once are also read-only.

Note

To work with an updatable query result set, you must have `psycopg2` driver version 2.8 or above installed.

Editable and read-only columns are identified using pencil and lock icons (respectively) in the column headers.

Query Tool Editable Columns

An updatable result set is similar to the Data Grid in View/Edit Data mode, and can be modified in the same way.

If Auto-commit is off, the data changes are made as part of the ongoing transaction, if no transaction is ongoing a new one is initiated. The data changes are not committed to the database unless the transaction is committed.

If any errors occur during saving (for example, trying to save NULL into a column with NOT NULL constraint) the data changes are rolled back to an automatically created SAVEPOINT to ensure any previously executed queries in the ongoing transaction are not rolled back.

All rowsets from previous queries or commands that are displayed in the *Data Output* panel will be discarded when you invoke another query; open another query tool browser tab to keep your previous results available.

Explain Tab

To generate the *Explain* or *Explain Analyze* plan of a query, click on *Explain* or *Explain Analyze* button in the toolbar.

More options related to *Explain* and *Explain Analyze* can be selected from the drop down on the right side of *Explain Analyze* button in the toolbar.

The Explain Options

Please note that pgAdmin generates the *Explain [Analyze]* plan in JSON format.

On successful generation of *Explain* plan, it will create three tabs/panels under the Explain panel.

Graphical Tab

Hover over an icon on the *Graphical* tab to review information about that item; a popup window will display information about the selected object. For information on JIT statistics, triggers and a summary, hover over the icon on top-right corner; a similar popup window will be displayed when appropriate.

Please note that *EXPLAIN VERBOSE* cannot be displayed graphically.

Use the download button on top left corner of the *Explain* canvas to download the plan as an SVG file.

> **Note:** Download as SVG is not supported on Internet Explorer.

A Graphical Explain

Note that the query plan that accompanies the *Explain analyze* is available on the *Data Output* tab.

Analysis Tab

The *Analysis* tab shows the plan details in table format, it generates a format similar to the format available at *explain.depsez.com*. Each row of the table represents the data for a *Explain Plan Node*. The output may contain the node information, exclusive timing, inclusive timing, actual vs. planned rows, actual rows, planned rows, or loops. When you select a row, the child rows of that selected row are marked with an orange dot.

If the percentage of the exclusive/inclusive timings of the total query time is:

Greater than 90 --> Red

Greater than 50 --> Orange (between red and yellow)

Greater than 10 --> Yellow

If the planner has misestimated the number of rows (actual vs planned) by:

10 times --> Yellow color

100 times --> Orange (between Red and Yellow) color

1000 times --> Red color

The Analysis Tab

Statistics Tab

The *Statistics* tab displays information in two tables:

- `Statistics per Node Type` tells you how many times each node type was referenced.
- `Statistics per Table` tells you how many times each table was referenced by the query.

The Statistics Tab

Messages Tab

Use the *Messages* tab to view information about the most recently executed query:

The Messages Tab

If the server returns an error, the error message will be displayed on the *Messages* tab, and the syntax that caused the error will be underlined in the SQL editor. If a query succeeds, the *Messages* tab displays how long the query took to complete and how many rows were retrieved:

A successful query

Notifications Tab

Use the *Notifications* tab to view details of the asynchronous notifications that a client process may have sent:

The Notifications Tab

You can see details such as recorded time of the asynchronous notification event, name of the event or channel, process ID of the client process that has sent the notification, and the payload string that might have been sent along with the notification.

Query History Panel

Use the *Query History* tab to review activity for the current session:

The Query History Tab

The Query History tab displays information about recent commands:

- The date and time that a query was invoked.
- The text of the query.
- The number of rows returned by the query.
- The amount of time it took the server to process the query and return a result set.
- Messages returned by the server (not noted on the *Messages* tab).
- The source of the query (indicated by icons corresponding to the toolbar).

You can show or hide the queries generated internally by pgAdmin (during 'View/Edit Data' or 'Save Data' operations).

To erase the content of the *Query History* tab, select *Clear history* from the *Clear* drop-down menu.

Query History is maintained across sessions for each database on a per-user basis when running in Query Tool mode. In View/Edit Data mode, history is not retained. By default, the last 20 queries are stored for each database. This can be adjusted in config_local.py by overriding the MAX_QUERY_HIST_STORED value.

Connection Status

Use the *Connection status* feature to view the current connection and transaction status by clicking on the status icon in query tool:

Checking the Connection Status

If the server gets disconnected, the connection icon at the top left corner turns red and a tool tip is displayed. The icon turns green once you restore the server connection.

The connection check failed

7.3 The PEM Schema Diff Tool

Schema Diff is a feature that allows you to compare schema objects between two database schemas. Use the `Tools` menu to access Schema Diff.

The Schema Diff feature allows you to:

- Compare and synchronize the database schemas (from source to target).
- Visualize the differences between database schemas.
- List the differences in SQL statement for target schema objects.
- Generate synchronization scripts.

Note - The source and target databases must be of the same major version.

Click on `Schema Diff` under the `Tools` menu to open a selection panel. Choose the source and target servers, databases, and schemas that will be compared. After selecting the objects, click on the `Compare` button.

You can open multiple copies of `Schema Diff` in individual tabs simultaneously. To close a copy of Schema Diff, click the `X` in the upper-right hand corner of the tab bar.

Schema Diff dialog

Use the `Preferences` dialog to specify if `Schema Diff` should open in a new browser tab. Set `Open in new browser tab` option to `true` .

The `Schema Diff` panel is divided into two panels; an `Object Comparison panel` and a `DDL Comparison panel` .

The Schema Diff Object Comparison Panel

In the object comparison panel, you can select the source and target servers of the same major version, databases, and schemas to be compared. You can select any server listed under the browser tree whether it is connected or disconnected. If you select a server that is not connected then it will prompt you for the password before using the server.

Next, select the databases that will be compared. The databases can be the same or different (and within the same server or from different servers).

Lastly, select the source and target schemas which will be compared.

Schema Diff dialog - Compare button

After you select servers, databases, and schemas, click on the `Compare` button to obtain the `Comparison Result`.

Schema Diff dialog - Comparison Results

Use the drop-down lists of Functions, Materialized Views, Tables, Trigger Functions, Procedures, and Views to view the DDL statements of all the schema objects.

In the upper-right hand corner of the object comparison panel is a `Filter` option that you can use to filter the schema objects based on the following comparison criteria:

- Identical – If the object is found in both schemas with the same SQL statement, then the comparison result is identical.
- Different – If the object is found in both schemas but have different SQL statements, then the comparison result is different.
- Source Only – If the object is found in source schema only and not in target schema, then the comparison result is source only.
- Target Only – If the object is found in target schema only and not in source schema, then the comparison result is target only.

Schema Diff dialog - Filter option

Click on any of the schema objects in the object comparison panel to display the DDL Statements of that object in the `DDL Comparison` panel.

Schema Diff DDL Comparison Panel

The `DDL Comparison` panel displays three columns:

- The first column displays the DDL statement of the object from the source schema.
- The second column displays the DDL statement of the object from the target schema.
- The third column displays the difference in the SQL statement of the target schema object.

Schema Diff dialog - DDL Comparison panel

You can review the DDL statements of all the schema objects to check for the differences in the SQL statements.

You can also use the Schema Diff tool to generate the SQL script of the differences found in the target schema object based on the SQL statement of the source schema object. To generate the script, select the checkboxes of the schema objects in the object comparison panel and then click on the `Generate Script` button in the upper-right hand corner of the object comparison panel.

Schema Diff dialog - Generate Script button

Select the schema objects and click on the `Generate Script` button to open the `Query Tool` in a new tab, with the difference in the SQL statement displayed in the `Query Editor`.

If you have clicked on the schema object to check the difference generated in the `DDL Comparison` Panel, and you have not selected the checkbox of the schema object, PEM will open the `Query Tool` in a new tab, with the differences in the SQL statements displayed in the `Query Editor`.

You can also use the `Copy` button to copy the difference generated in the `DDL Comparison` panel.

Schema Diff dialog - Generate Script - Query Editor

Apply the SQL Statement in the target schema to synchronize the schemas.

7.4 Performance Monitoring and Management

Performance Monitoring and Management

PEM contains built-in functionality that implements enterprise-wide performance monitoring of all managed servers. While you can customize many aspects of the various performance monitoring aspects of PEM, you can also elect to accept the recommended defaults that come out-of-the-box with the product.

The Global Overview dashboard

The top-level dashboard is the `Global Overview` . The Global Overview presents a status summary of all the servers and agents that are being monitored by the PEM server, a list of the monitored servers, and the state of any currently triggered alerts.

Using Dashboards to View Performance Information

Using Dashboards to View Performance Information

PEM displays performance statistics through a number of dashboards; each dashboard contains a series of summary views that contain charts, graphs and tables that display the statistics related to the selected object.

The PEM client displays the Global Overview dashboard when it connects to the PEM server. Additional dashboards provide statistical information about monitored objects. These include the:

Alerts Dashboard

The Alerts dashboard displays the currently triggered alerts. If opened from the Global Overview, the dashboard displays the current alerts for all monitored nodes on the system; if opened from a node within a server, the report will reflect alerts related to that node, and all monitored objects that reside below that object in the tree control.

Audit Log Analysis dashboard

For Advanced Server users, the Audit Log Analysis dashboard allows you to browse the audit logs that have been collected from instances that have audit logging and collection enabled.

Database Analysis dashboard

The Database Analysis dashboard displays performance statistics for the selected database.

I/O Analysis dashboard

The I/O Analysis dashboard displays I/O activity across various areas such as object DML activity, log operations and more.

Memory Analysis dashboard

The Memory Analysis dashboard supplies statistics concerning various memory-related metrics for the Postgres server.

Object Activity Analysis dashboard

The Object Activity Analysis dashboard provides performance details on tables/indexes of a selected database.

Operating System Analysis dashboard

The Operating System Analysis dashboard supplies information regarding the performance of the underlying machine’s operating system.

Probe Log Analysis Dashboard

The Probe Log Analysis dashboard displays any error messages returned by a PEM agent.

Server Analysis dashboard

The Server Analysis dashboard provides general performance information about the overall operations of a selected Postgres server.

Server Log Analysis dashboard

The Server Log Analysis dashboard allows you to filter and review the contents of server logs that are stored on the PEM server.

Session Activity Analysis dashboard

The Session Activity Analysis dashboard provides information about the session workload and lock activity for the selected server

Session Waits Analysis dashboard

The Session Waits Analysis dashboard provides an overview of the current DRITA wait events for an Advanced Server session.

Storage Analysis dashboard

The Storage Analysis dashboard displays space-related metrics for tablespaces and objects.

System Waits Analysis dashboard

The System Waits Analysis dashboard displays a graphical analysis of system wait information for an Advanced Server session.

Streaming Replication Analysis dashboard

The Streaming Replication Analysis dashboard displays statistical information about WAL activity for a monitored server and allows you to monitor the status of Failover Manager clusters.

There are two ways to open a dashboard; you can:

- Select an active dashboard name from the Dashboards menu (accessed via the Management menu).
- Right click on the name of a monitored object in the tree control and select the name of the dashboard you would like to review from the Dashboards menu.

Each dashboard is displayed on the `Monitoring` tab in the main panel of the client window. After opening a dashboard, you can navigate to other dashboards within the same tab.

Each dashboard header includes navigation menus that allow you to navigate to other dashboards; use your browsers forward and back icons to scroll through previously-viewed dashboards. Use the Refresh icon to update the current dashboard.

Options on the `Dashboard Configuration` dialog allow you to link the time lines of all of the line graphs on the dashboard. To open the `Dashboard Configuration` dialog, click the Settings icon displayed in the dashboard header.

The Dashboard Configuration dialog

Use fields on the `Dashboard Configuration` dialog to control attributes of the charts displayed on the dashboard:

- Set the Link timelines of all the line charts slider to Enable to indicate that the specified timeline should be applied to line graphs displayed on the dashboard; if set to Disable, your preferences will be preserved for later use, but will not modify the amount of data displayed.
- Use the `Days` selector to specify the number of days of gathered data that should be displayed on line graphs.
- Use the `Hour(s)` selector to specify the number of hours of gathered data that should be displayed on line graphs.
- Check the box next to Remember configuration for this dashboard to indicate that the customized time span should be applied to the current dashboard only; if left unchecked, the time span will be applied globally to line graphs on all dashboards.

Please note that settings specified on the Dashboard Configuration dialog are applied only to the current user's session.

Managing Custom Dashboards

Managing Custom Dashboards

PEM displays performance statistics through a number of system-defined dashboards; each dashboard contains a series of summary views that contain charts, graphs and tables that display statistics related to the selected object. You can use the `Manage Dashboards` tab to create and manage custom dashboards that display the information that is most relevant to your system.

The Manage Dashboards tab

To create a custom dashboard, click the `Create New Dashboard` link (located in the Quick Links section of the Manage Dashboards tab).

To modify an existing dashboard, click the edit icon to the left of a dashboard name. The dashboard editor will open, displaying the definition of the dashboard. When you've finished modifying the dashboard's definition, click the Save button to preserve your changes; click Cancel to exit without saving your changes.

To delete a dashboard, click the delete icon to the left of a dashboard name. A popup will ask you to confirm that you wish to delete the dashboard; click OK to delete the selected dashboard.

Creating a Custom Dashboard

You can use the PEM dashboard editor to create or modify a user-defined dashboard. The custom dashboard may include pre-defined charts, user-defined charts or a mix of pre-defined and user-defined charts.

The Create Dashboard editor

Use the fields in the `Configure` section to specify general information about the dashboard:

- Specify a name for the dashboard in the `Name` field. The name specified will also be the title of the dashboard if the title is displayed.
- Use the `Level` drop-down listbox to specify the level of the PEM hierarchy within the PEM client on which the dashboard will be displayed. A dashboard may be accessed via the Dashboards menu on a Global level, an Agent level, the Server level or the Database level. Each selected level within the list will expose a different set of metrics on which the custom dashboard's charts may be based.
- Provide a description of the dashboard in the Description field.

Provide information in the fields in the `Ops dashboard options` box if the dashboard will be used as an Ops dashboard:

- Set the `Ops Dashboard?` field to Yes to instruct the server to create a dashboard that is formatted for display on an Ops monitor.
- Set the `Show Title?` field to Yes to display the dashboard name at the top of the Ops dashboard.
- Use the `Font` drop-down list box to select a custom font style for the title. The selected font style will be displayed in the Preview box.
- Use the `Font size` drop-down list box to select a custom font size for the title. The selected font style will be displayed in the Preview box.

Use the `Permissions` box to specify the users that will be able to view the new dashboard:

- Set the `Share with all slider` to Yes to instruct the server to allow all Teams to access the dashboard, or set Share with all to No to enable the Access permissions field.
- Use the `Access permissions` field to specify which roles can view the new dashboard. Click in the field, and select from the list of users to add a role to the list of users with dashboard access.

When you've completed the Configure Dashboard section, click the arrow in the upper-right corner to close the section, and access the Dashboard Layout Design section.

Modifying a Section Header

Click the edit icon in a section header to specify a section name; then, click the add icon (+) to add a chart to the section.

Adding a Chart

Use the arrows to the right of each chart category to display the charts available and select a chart.

Specifying placement details for a chart

Use the chart detail selectors to specify placement details for the chart:

- Use the `Chart width` selector to indicate the width of the chart; select 50% to display the chart in half of the dashboard, or 100% to use the whole dashboard width.

- Use the `Chart alignment` selector to indicate the position of the chart within the section:

  Select `Left` to indicate that the chart should be left-justified.

  Select `Center` to indicate that the chart should be centered.

Select `Right` to indicate that the chart should be right-justified.

Please note that tables are always displayed centered.

When creating or editing a custom dashboard, you can use drag and drop to re-arrange the charts within a section or to move a chart to a different section.

To add another chart to your dashboard, click the add icon (+) in the section header. When you've finished editing the dashboard, click the Save button to save your edits and exit.

To exit without saving your changes, click the `Cancel` button.

Creating an Ops Dashboard

You can use the PEM dashboard editor to create a custom dashboard formatted for display on an Ops monitor. An Ops dashboard displays the specified charts and graphs, while omitting header information and minimizing extra banners, titles, and borders.

Ops dashboard options

To create an `Ops dashboard`, provide detailed information about the Ops display in the Ops dashboard options section of the `Create Dashboard` dialog.

- Set the `Ops Dashboard?` field to Yes to instruct the server to create a dashboard that is formatted for display on an Ops monitor.
- Set the `Show Title?` field to Yes to display the dashboard name at the top of the Ops dashboard.
- Use the `Font` drop-down list box to select a custom font style for the title. The selected font style will be displayed in the Preview box.
- Use the `Font size` drop-down list box to select a custom font size for the title. The selected font style will be displayed in the Preview box.

After adding charts and tables to the Ops dashboard, click the Save button to save your work. You can then access the dashboard by navigating through the Dashboards menu of the hierarchy level specified in the Level field on the New Dashboard dialog.

Using the Manage Charts tab

You can use the `Manage Charts` tab to access dialogs that allow you to create or modify a custom line chart or table, or import a Capacity Manager template for use in a custom chart. After defining a chart, you can display the chart on a custom dashboard. To open the `Manage Charts` tab, select `Manage Charts...` from the PEM client `Management` menu.

The Manage Charts tab

The Manage Charts tab provides a Quick Links menu that allows you to access dialogs to:

- Create a New Chart for use on a custom dashboard.
- Import a Capacity Manager template to use as a template for creating a custom chart.

The `Custom Charts` table displays a list of user-defined charts; when a chart is newly added, the font displays in green. When you add an additional chart or refresh the screen, the name of the chart is displayed in black.

The Custom Charts table

Use the search box in the upper-right hand corner of the Custom Charts table to search through your custom charts. Specify a:

- Chart name
- Type
- Level
- Metrics Category

Use icons to the left of a charts name in the `Custom Charts` table to manage a chart:

- Click the edit icon to open the `Chart Configuration` wizard and modify aspects of the chart or table.

- Click the delete icon to delete the selected chart.

Creating a Custom Chart

Click the `Create New Chart` icon in the `Quick Links` section of the `Manage Charts` tab to open the `Create Chart` wizard. The wizard will walk you through the steps required to define a new chart.

Specifying general information about the chart

Use the fields on the `Configure Chart` dialog to specify general information about the chart:

- Specify the name of the chart in the `Name` field.
- Use the drop-down listbox in the `Category` field to specify the category in which this chart will be displayed; when adding a custom chart to a custom dashboard, the chart will be displayed for selection in the category specified.
- Use the radio buttons in the `Type` field to specify if the chart will be a `Line chart` or a `Table` .
- Provide a description of the chart in the `Description` field. The description will be displayed to the user viewing the chart (on a custom dashboard) when they click the information icon.

When you've completed the fields on the `Configure Chart` dialog, click Next to continue.

Specifying the metrics that will be displayed

Use the fields on the `Select Metrics` dialog to select the metrics that will be displayed on the chart.

- Use the `Metric level` drop-down listbox to specify the level of the PEM hierarchy from which you wish to select metrics. You can specify Agent, Database, or Server. Each level offers access to a unique set of probes and metrics.
- Use the tree control in the Available metrics box to select the metrics that will be displayed on the chart.

  If you are creating a table, you may only select metrics from one probe; each node of the tree control lists the metrics returned by a single probe. Expand a node of the tree control, and check the boxes to the left of a metric name to include that metric data in the table.

  If you are creating a line chart, expand the nodes of the tree control and double-click each metric that you would like to include in the chart.

- Use the fields in the Selected metrics panel to specify how the metric data will be displayed in your chart. The selection panel displays the name of the metric in the (non-modifiable) Metric [Probe] column. You can:
  - Click the garbage can icon to delete a metric from the list of selected metrics.
  - Use the drop-down listboxes in the `Selection Criteria` column to specify the order of the data displayed.
  - Use the `Limit` field to specify the number of rows in a table or lines in a chart:

  The maximum number of lines allowed in a chart is 32.

  The maximum number of rows allowed in a table is 100.

- If you are creating a line chart, PEM supports comparisons of cross-hierarchy metrics.
  - Click the `compare icon` to open a selection box that allows you to select one or more probe-specific attributes (i.e. CPUs, interfaces, databases, etc.) to compare in the chart.
  - Click the `copy` icon to apply your selections to all of the metrics for the same probe. When the popup opens, click Yes to confirm that other selections for the same probe will be overwritten, or No to exit the popup without copying the attributes.

When you've completed the fields on the `Select Metrics` dialog, click Next to continue.

Specifying chart options

Use the fields on the `Set Options` dialog to specify display options for your chart:

- Use the `Auto Refresh` field to specify the number of minutes between chart updates - choose a value from 1 to 120. The default auto refresh rate is 2 minutes.

Use fields under the Line chart options heading to specify display preferences for a line chart:

- Use the `Points to plot` field to specify the maximum number of points that will be plotted on the chart.
- Use the fields to the right of the Historical span label to specify how much historical data should be displayed on the chart:

  Use the `Day(s)` field to specify the number of days of historical data that should be included on the chart.

  Use the `Hour(s)` field to specify the number of hours of historical data that should be included on the chart.

  Use the `Minute(s)` field to specify the number of minutes of historical data that should be included on the chart.

Use the fields in the `Data extrapolation` box to specify if PEM should generate extrapolated data based on historical data:

- Click the `No Extrapolation` label to omit extrapolated data from the chart.
- Click the `Span` label to use the Days and Hours selectors to specify the period of time spanned by the metrics on the chart.
- Click the `Threshold` label to use threshold selectors to specify a maximum or minimum value for the chart.

When you've completed the fields on the Set Options dialog, click Next to continue.

Specifying access permissions

Use the fields on the `Set Permissions` dialog to specify display options for your chart.

- Set the `Share with all slider` to Yes to indicate that the chart will be available to all authorized users, or No to restrict access to the users or groups specified in the Access permissions field.
- Use the `Access permissions` field to select the group or groups that will have access to the chart.

The chart definition is displayed on the Manage Charts tab

When you've finished defining the chart, click `Finish` to save your edits and add your chart to the list on the `Manage Charts` tab.

Importing a Capacity Manager Template

Importing a Capacity Manager Template

Click the `Import Capacity Manager Template` icon in the Quick Links section of the `Manage Charts` tab to open the `Create Chart` dialog, and use a Capacity Manager template as a starting point for a chart or table.

Importing a Capacity Manager template

When the `Create Chart` dialog opens, provide information about the custom chart:

- Use the drop-down listbox in the `Import capacity template` field to select the name of the template on which the chart will be based.
- Specify the name of the chart in the `Name` field.
- Use the drop-down listbox in the `Category` field to specify the category in which this chart will be displayed. When adding a custom chart to a custom dashboard, the chart will be displayed for selection in the Category specified.
- Use the radio buttons in the `Type` field to specify if the chart will be a `Line chart` or a `Table` .
- Provide a description of the chart in the `Description` field. The description will be displayed to the user viewing the chart (on a custom dashboard) when they click the information icon.

Click Next to continue to the Select Metrics dialog.

The template metrics

The `Select Metrics` window allows you to review the metrics specified by the selected template. The bottom panel of the chart editor displays the metrics that will be included in the chart. The metrics included in the chart are not modifiable via the chart editor; to modify the metrics, you must use the Capacity Manager utility to update the template.

When you've reviewed the metrics, click Next to continue to the Set Options dialog.

Selecting chart options

Use the fields on the `Set Options` window to specify display options for your chart:

- Use the `Auto Refresh` field to specify the number of minutes between chart updates - choose a value from 1 to 120. The default auto refresh rate is 2 minutes.

Use the fields in the `Data extrapolation` box to specify the time period covered by the chart. You can either:

- click the `Historical days and extrapolated days` label and provide:
  - the number of days of historical data that should be charted in the `Historical` field.
  - the number of projected days that should be charted in the `Extrapolated` field.
- or, click the Historical days and threshold label and provide:
  - the number of days of historical data that should be charted in the `Historical` field
  - the `threshold` value at which the chart will end.

When you've completed the Set Options window, click `Next` to continue.

Selecting permissions for the chart

Use the fields on the `Set Permissions` window to specify display options for your chart:

- Set the `Share with all slider` to Yes to indicate that the chart will be available to all authorized users, or No to restrict access to the users or groups specified in the Access permissions field.
- Use the `Access permissions` field to select the group or groups that will have access to the chart.

When you've finished defining the chart, click `Finish` to save your edits and add your chart to the list on the `Manage Charts` tab.

Customizing Probes

Customizing Probes

A probe is a scheduled task that returns a set of performance metrics about a specific monitored object. A probe retrieves statistics from a monitored server, database, operating system or agent. You can use the Manage Probes tab (shown in Figure 6.13) to override the default configuration and customize the behavior of each probe.

To open the `Manage Probes` tab, select `Manage Probes...` from the `Management` menu. The Manage Probes tab opens in the PEM client.

The Manage Probes tab

The `Manage Probes` tab provides a set of Quick Links that you can use to create and manage probes:

- Click the `Manage Custom Probes` icon to open the `Custom Probes` tab and create or modify a custom probe.
- Click the `Copy Probes` icon to open the Copy Probe dialog, and copy the probe configurations from the currently selected object to one or more monitored objects.

A probe monitors a unique set of metrics for each specific object type (server, database, database object, or agent); select the name of an object in the tree control to review the probes for that object.

To modify the properties associated with a probe, highlight the name of a probe, and customize the settings that are displayed in the Probes table:

- Move the `Default` switch in the `Execution Frequency` columns to `N` to enable the Minutes and Seconds selectors, and specify a non-default value for the length of time between executions of the probe.
- Move the `Default` switch in the `Enabled?` column to `No` to change the state of the probe, and indicate if the probe is active or not active.

Note

If data from a Disabled probe is used in a chart, the chart will display an information icon in the upper-left corner that allows you to enable the probe by clicking the provided link.

- Move the `Default` switch in the `Data Retention` column to No to enable the Day(s) field and specify the number of days that information gathered by the probe is stored on the PEM server.

The `Manage Probes` tab may display information about probes that cannot be modified from the current node. If a probe cannot be modified from the current dialog, the switches are disabled. Generally, a disabled probe can be modified from a node that is higher in the hierarchy of the PEM client tree control; select another object in the tree control to modify which probes are displayed or enabled in the `Manage Probes` tab.

Creating a Custom Probe

Creating a Custom Probe

You can use the `PEM Custom Probes` tab to create a new probe or modify an existing user-defined probe. To open the `Custom Probes` tab, select the `Manage Custom Probes...` icon from the `Manage Probes` tab.

The Custom Probes dialog

Use the `Show System Probes?` switch to display or conceal the system probes on the Custom Probes tab.

You can use the `Custom Probes` tab to create a new probe or modify an existing probe. To create a new probe, click the Add icon in the upper-right corner of the tab; provide a name for the new probe in the Probe Name column. Then, select the Edit icon (located to the left of the probe name) to review or add the probe definition.

Defining a custom probe – the General tab

Use the fields on the `General` tab to modify the definition of an existing probe or to specify the properties of a new probe:

- Use the `Probe Name` field to provide a name for a new probe.
- Use the `Collection method` field to specify the probe type. Use the drop-down listbox to select:
- SQL - the probe will gather information via a SQL statement.
- WMI - the probe will gather information via a Windows Management Instrumentation extension.
- Batch - the probe will use a command-script or shell-script to gather information.

  Before creating a batch probe on a Linux system, you must modify the agent.cfg file, setting the allow_batch_probes parameter equal to true and restart the PEM agent. The `agent.cfg` file is located in one of the following directories:

  - If you have installed PEM using graphical installer: `/opt/edb/pem/agent/etc/agent.cfg`
  - If you have installed PEM using RPM: `/usr/edb/pem/agent/etc/agent.cfg`

  On 64-bit Windows systems, agent settings are stored in the registry. Before creating a batch probe, modify the registry entry for the `AllowBatchProbes` registry entry and restart the PEM agent. PEM registry entries are located in `HKEY_LOCAL_MACHINE\Software\Wow6432Node\EnterpriseDB\PEM\agent` .

Please note that batch probes are platform-specific. If you specify a collection method of Batch, you must specify a platform type in the Platform field.

To invoke a script on a Linux system, you must modify the entry for `batch_script_user` parameter of `agent.cfg` file and specify the user that should be used to run the script. You can either specify a non-root user or root for this parameter. If you do not specify a user, or the specified user does not exist, then the script will not be executed. Restart the agent after modifying the file.

To invoke a script on a Windows system, set the registry entry for `AllowBatchJobSteps` to true and restart the PEM agent.

- Use the `Target Type` drop-down listbox to select the object type that the probe will monitor. Target type is disabled if Collection method is WMI.
- Use the `Minutes` and `Seconds` selectors to specify how often the probe will collect data.
- Use the `Probe Enable?` switch to specify if the probe in enabled by default. Specify Yes to enable the probe by default, or No to specify that the probe is disabled by default.

Note

If data from a disabled probe is used in a chart, the chart will display an information icon in the upper-left corner that allows you to enable the probe by clicking the provided link.

- Use the `Data Retention` field to specify the number of days that gathered information will be retained in the probe's history table.
- Use the switch next to `Discard from history` to specify if the server should create a history table for the probe. Select Yes to discard probe history, or No to retain the probe history in a table.
- Use the `Platform` drop-down listbox to specify the type of platform that the probe will monitor. This field is enabled only when the Collection method is Batch.

The Columns tab of the Custom Probes dialog

Use the `Columns` tab to define the columns in which the probe data will be stored. Navigate to the `Columns` tab, and click the Add button (in the upper-right corner) to define a new column. After a providing a column name in the Name field, click the Edit button (to the left of the new column name) to provide information about the column:

- Provide a descriptive name for the column in the `Name` field.
- The `Internal Name` field is not enabled for user-defined probes.
- Use the `Column Type` drop-down listbox to specify if the column is a Key column (a primary key) or a Non key column. Non-key columns are generally metric items (values that can be graphed).
- Use the `Data Type` drop-down listbox to specify the type of data that will be stored in the column.
- Use the `Unit` field to specify the unit of measure that applies to the metric stored in the column. This unit is displayed on the Y-Axis of a custom chart or a Capacity Manager chart. This is an optional field.
- Use the `Graphable` switch to specify if the defined metric may be graphed, and that the probe should be accessible from the Capacity Manager or Manage Charts dialogs.
- Use the `Is PIT` switch to specify if the metric should be stored by point-in-time.

  'Point-in-time' metrics are those metrics that change (increase or decrease) at any given point of time. For example, database size is a point-in-time metric; at any given point-in-time, the size of the database is fluctuating. Metrics that are not point-in-time (also referred to as cumulative metrics) are metrics whose size always increases over time. For example, Blocks Read and Tuples Read are cumulative metrics; the value stays the same or increases.
- Use the `Calculate PIT` switch to specify that the server should calculate a point-in-time value for the metric data. Calculate PIT is disabled if Is PIT is Yes.

  PEM allows you to store point-in time-values of cumulative metrics as well. PEM subtracts the last collected value of a cumulative metric from the current value, and stores the difference as a point-in-time value.

The Code tab of the Custom Probes dialog

Use the `Code` tab to specify the default code that will be executed by the probe:

- If the probe is a SQL probe, you must specify the `SQL SELECT` statement invoked by the probe on the `Code` tab. The column names returned by the query must match the Internal Name specified on the `Columns` tab. The number of columns returned by the query, as well as the column name, data type, etc. must match the information specified on the `Columns` tab.
- If the probe is a batch probe, you must specify the shell or .bat script that will be invoked when the probe runs. The output of the script should be as follows:

  The first line must contain the names of the columns provided on the `Columns` tab. Each column name should be separated by a tab (t) character. From the second line onwards, each line should contain the data for each column, separated by a tab character.

  If a specified column is defined as key column, you should ensure that the script does not produce duplicate data for that column across lines of output. The number of columns specified in the `Columns` tab and their names, data type, etc. should match with the output of the script output.
- If the probe is a WMI probe, you must specify the WMI query as a `SELECT WMI` query. The column name referenced in the `SELECT` statement should be same as the name of the corresponding column specified on the `Column` tab. The column names returned by the query must match the `Internal Name` specified on the `Column` tab. The number of columns returned by the query, as well as the column name, data type, etc. must match the information specified on the Columns tab.

The Alternate Code tab of the Custom Probes dialog

Use the `Alternate Code` tab to provide code that will be invoked if the probe fires on a specific version of the server. To provide version-specific code, move the `Applies to any server version?` switch to `No` , and click the `Add` button. Then, use the `Database Version(s)` drop-down listbox to select a version, and click the `Edit` button (to the left of the version name) to provide the code that will execute when the probe fires.

If you select a database version, and leave the `Probe Code` column blank, PEM will invoke the code specified on the `Code` tab when the probe executes on a server that matches that version.

When you've finished defining the probe, click the `Save` icon (in the corner of the `Custom Probes` tab) to save the definition, and make the probe data available for use on custom charts and graphs.

Deleting a Probe

Deleting a Probe

Use the Delete icon (located to the left of a Probe Name) to delete a user-defined probe. When you delete a probe, the probe is marked for deletion and will be deleted later (when custom probes are purged). During the deletion, the probe definition is deleted and any corresponding tables are dropped from the pemdata and pemhistory schemas.

System probes are the built-in probes provided by PEM, and are part of the PEM schema. If you attempt to delete a system probe, the PEM client will display a notice, informing you that the probe cannot be deleted.

Attempting to delete a system probe

Copying a Probe

Copying a Probe

You can use the `Copy Probe Configuration...` dialog to copy probe definitions from one monitored object to one or more monitored objects of the same type. To open the `Copy Probe Configuration...` dialog, highlight the object from which you are copying probes in the PEM client tree control, and select

`Manage Probes` from the `Management` menu. When the `Manage Probes` tab opens, click on `Copy Probe` to open the `Copy Probe Configuration` dialog:

The Copy Probe Configuration tree control

The dialog will copy the probe definitions from the object through which the Copy Probe Configuration dialog was opened, to the location(s) selected on the tree control.

Note that if you specify a parent node in the Copy Probe Configuration tree control, PEM will copy the probe configurations to each object (of the same type) that resides under that node in the tree control. For example, to copy the probe definitions from one schema to all schemas that reside within a database, select only the parent database of the target schemas. Please note that a red warning symbol is displayed to the left of the name of a listed target object if that object is the source of the probe that is being copied.

When you have selected the target object or objects, click the `Configure Probes` button to copy the probe definitions to the location selected on the dialog.

Alerting

Alerting

PEM continually monitors registered servers and compares performance metrics against pre-defined and user-specified thresholds that constitute good or acceptable performance for each statistic. Any deviation from an acceptable threshold value triggers an alert. An alert is a system-defined or user-defined set of conditions that PEM compares to the system statistics. Alerts call your attention to conditions on registered servers that require your attention.

Reviewing alerts on the Global Overview

When your system statistics deviate from the boundaries specified for that statistic, the alert triggers, displaying a high (red), low (yellow), or medium (orange) severity warning in the left-most column of the `Alert Status` table on the `Global Overview` dashboard.

The Alert Status table

The PEM server includes a number of pre-defined alerts that are actively monitoring your servers. If the alert definition makes details available about the cause of the alert, you can click the down arrow to the right of the severity warning to access a dialog with detailed information about the condition that triggered the alert.

Alert details

PEM also provides an interface that allows you to create customized alerts. Each alert uses metrics defined on an alert template. An alert template defines how the server will evaluate the statistics for a resource or metric. The PEM server includes a number of pre-defined alert templates, or you can create custom alert templates.

Using the Alerts Dashboard

Using the Alerts Dashboard

Use the `Dashboards` menu (on the `Monitoring` tab) to access the `Alerts` Dashboard. The `Alerts` Dashboard displays a summary of the active alerts and the status of each alert:

The Alerts Dashboard

The `Alerts Dashboard` header displays the date and time that the dashboard was last updated, and the number of current alerts.

The `Alerts Overview` section displays a graphic representation of the active alerts, as well as a count of the current high, low and medium alerts. The vertical bar on the left of the graph provides the count of the alerts displayed in each column. Hover over a bar to display the alert count for the selected alert severity in the upper-right hand corner of the graph.

The `Alert Details` table provides a list of the alerts that are currently triggered. The entries are prioritized from high-severity to lower-severity; each entry includes information that will allow you to identify the alert and recognize the condition that triggered the alert. Click the name of an alert to review detailed information about the alert definition.

The `Alert Errors` table displays configuration-related errors (eg. accidentally disabling a required probe, or improperly configuring an alert parameter). You can use the information provided in the Error Message column to identify and resolve the conflict that is causing the error.

Customizing the Alerts Dashboard

You can customize tables and charts that appear on the Alerts dashboard. To customize a table or chart, click the Settings icon located in the upper-right corner.

Customizing a chart

Use fields on the Personalize chart configuration dialog (Figure 4.32) to provide your display preferences:

- Use the `Auto Refresh` field to specify the number of seconds between updates of the data displayed in the table or chart.
- If applicable, use the `Download as` field to indicate if you would like a chart to be downloaded as a JPEG image or a PNG image.
- If applicable, use the `Colours selectors` to specify the display colors that will be used on a chart.
- If applicable, set the `Show Acknowledged Alerts` switch to Yes indicate that you would like the table to display alerts that you have acknowledged with a checkbox in the Ack'ed column. Set the field to No to indicate that the table should hide any acknowledged alerts. The switch acts as a toggle; acknowledged alerts are not purged from the table content until the time specified in the alert definition passes.

To save your customizations, click the `Save` icon (a check mark) in the upper-right corner; to delete any previous changes and revert to the default values, click the `Delete` icon. The `Save` and `Delete` drop-down menus allow you to specify if your preferences should be applied to `All Dashboards` , or to a selected server or database.

Using the Manage Alerts Tab

Using the Manage Alerts Tab

Use the PEM Client's `Manage Alerts` tab to define, copy, or manage alerts. To open the `Manage Alerts` tab, select `Manage Alerts` from the `Management` menu.

The Manage Alerts tab

Use the `Quick Links` toolbar to open dialogs and tabs that will assist you when managing alerts:

- Click `Copy Alerts` to open the `Copy Alert Configuration` dialog and copy an alert definition.
- Click `Alert Templates` to open the `Alert Template` tab, and modify or create an alert template.
- Click `Email Groups` to open the `Email Groups` tab, and modify or create an email group.
- Click `Server Configurations` to open the `Server Configuration` dialog and review or modify server configuration settings.
- Click `Help` to open the PEM online help in a new tab of the PEM web interface.

Use the table in the `Alerts` section of the `Manage Alerts` tab to create new alerts or manage existing alerts.

Creating a Custom Alert Template An alert template is a prototype that defines the properties of an alert. An alert instructs the server to compare the current state of the monitored object to a threshold (specified in the alert template) to determine if a situation exists that requires administrative attention.

You can use the `Alert Templates` tab to define a custom alert template or view the definitions of existing alert templates. To open the `Alert Templates` tab, select the `Manage Alerts...` menu option from the `Management` menu. When the Manage Alerts tab opens, select Alert Templates from the Quick Links toolbar.

The Alert Templates tab

Use the `Show System Template` drop-down listbox to filter the alert templates that are displayed in the Alert Templates table. Use the listbox to select a level of the PEM hierarchy to view all of the templates for the selected level.

Defining a New Alert Template

To define a new alert template, use the `Show System Template` drop-down listbox to select None, and click the Add icon (+) located in the upper-right corner of the alert template table. The alert template editor opens.

The General tab

Use fields on the `General` tab to specify general information about the template:

- Use the `Template name` field to specify a name for the new alert template.
- Use the `Description` field to provide a description of the alert template.
- Use the `Target type` drop-down listbox to select the type of object that will be the focus of the alert.
- Use the `Applies to server` drop-down listbox to specify the server type (EDB Postgres Advanced Server or PostgreSQL) to which the alert will be applied; you can specify a single server type, or ALL.
- Use the `History retention` field to specify the number of days that the result of the alert execution will be stored on the PEM server.
- Use the `Threshold unit` field to specify the unit type of the threshold value.
- Use fields in the `Auto create` box to indicate if PEM should use the template to generate an automatic alert. If enabled, PEM will automatically create an alert when a new server or agent (as specified by the Target type drop-down listbox) is added, and delete that alert when the target object is dropped.
  - Move the `Auto create?` slider to `Yes` to indicate that PEM should automatically create alerts based on the template. If you modify an existing alert template, changing the Auto create? slider from No to Yes, PEM will create alerts on the existing agents and servers. Please note that if you change the slider from Yes to No, the default threshold values in existing alerts will be erased, and cannot be recovered.
  - Use the `Operator` drop-down listbox to select the operator that PEM will use when evaluating the current system values.

  Select a greater-than sign (>) to indicate that the alert should be triggered when the system values are greater than the values entered in the Threshold values fields.

  Select a less-than sign (<) to indicate that the alert should be triggered when the system values are less than the values entered in the Threshold values fields.

- Use the threshold fields to specify the values that PEM will compare to the system values to determine if an alert should be raised. Please note that you must specify values for all three thresholds (Low, Medium, and High):

  Enter a value that will trigger a low-severity alert in the `Low` field.

  Enter a value that will trigger a medium-severity alert in the `Medium` field.

  Enter a value that will trigger a high-severity alert in the `High` field.

- Use the `Check frequency` field to specify the default number of minutes between alert executions. This value specifies how often the server will invoke the SQL code specified in the definition and compare the result to the threshold value specified in the template.

The Probe Dependency tab of the Alert Templates dialog

Use the fields on the `Probe Dependency` tab to specify the names of probes referred to in the SQL query specified on the SQL tab:

- Use the `Probes` drop-down listbox to select from a list of the available probes; highlight a probe name, and click the Add button to add the probe to the list of probes used by the alert template. To remove a probe from the selected probes list, highlight the probe name, and click the Delete icon.

The Parameters tab of the Alert Templates dialog

- Use fields on the `Parameters` tab to define the parameters that will be used in the SQL code specified on the `SQL` tab. Click the `Add` icon (+) and:

  Use the `Name` field to specify the parameter name.

  Use the `Data type` drop-down listbox to specify the type of parameter.

  Use the `Unit` field to specify the type of unit specified by the parameter.

- Use the `Code` field on the `SQL` tab to provide the text of the SQL query that the server will invoke when executing the alert. The SQL query will provide the result against which the threshold value is compared; if the alert result deviates from the specified threshold value, an alert will be raised.

The SQL tab of the Alert Templates dialog

Within the query, parameters defined on the `Parameters` tab should be referenced sequentially by the variable param*x*, where *x* indicates the position of the parameter definition within the parameter list. For example, param_1 refers to the first parameter in the parameter list, param_2 refers to the second parameter in the parameter list, and so on.

The query can also include the following pre-defined variables:

Variable Description	Variable Name
agent identifier	'${agent_id}'
server identifier	'${server_id}'
database name	'${database_name}'
schema name	'${schema_name}'
Table	'${object_name}'
index	'${object_name}'
sequence	'${object_name}'
function name	'${object_name}'

- Use the `Detailed Information SQL` field to provide a SQL query that will be invoked if the alert is triggered. The result set of the query may be displayed as part of the detailed alert information on the Alerts dashboard or Global Overview dashboard.

Note

If the specified query is dependent on one or more probes from different levels within the PEM hierarchy (server, database, schema, etc.), and a probe becomes disabled, any resulting alerts will be displayed as follows:

- If the alert definition and the probe referenced by the query are from the same level within the PEM hierarchy, the server will display any alerts that reference the alert template on the `Alert Error` table of the `Global Alert Dashboard` .
- If the alert definition and the probe referenced by the query are from different levels of the PEM hierarchy, the server will display any triggered alerts that reference the alert template on the `Alert Details` table of the hierarchy on which the alert was defined.

Click the Save icon to save the alert template definition and add the template name to the Alert Templates list. After saving a custom alert template, you can use the Alerting dialog to define an alert based on the template.

Modifying or Deleting an Alert Template

To view the definition of an existing template (including PEM pre-defined alert templates), use the `Show System Template` drop-down listbox to select the type of object monitored. When you select the object type, the `Alert Templates` table will display the currently defined alert templates that correspond with that object type.

Highlight a Template Name in the list, and click the Edit icon (at the left end of the row) to review the template definition.

Use the tabs on the `Alert Templates` dialog to view detailed information about the alert template:

- General information is displayed on the `General` tab.
- The names of probes that provide data for the template are listed on the `Probe Dependency` tab.
- The names of any parameters referred to in the SQL code are listed on the `Parameters` tab.
- The SQL code that defines the behavior of the alert is displayed on the `SQL` tab.

To delete an alert template, highlight the template name in the alert templates table, and click the Delete icon. The alert history will persist for the length of time specified in the `History Retention` field in the template definition.

Creating a New Alert The `Manage Alerts` tab displays a table of alerts that are defined on the object currently selected in the PEM client tree control. You can use the `Alerts` table to modify an existing alert, or to create a new alert.

The Manage Alerts tab

To open the alert editor and create a new alert, click the Add icon (+) in the upper-right corner of the table. The editor opens as shown below.

The General tab of the alert editor

Use the fields on the `General` tab to provide information about the alert:

- Enter the name of the alert in the `Name` field.
- Use the drop-down listbox in the `Template` field to select a template for the alert. An alert template is a function that uses one (or more) metrics or parameters to generate a value to which PEM compares user-specified alert boundaries. If the value returned by the template function evaluates to a value that is within the boundary of a user-defined alert (as specified by the Operator and Threshold values fields), PEM raises an alert, adds a notice to the Alerts overview display, and performs any actions specified on the template.
- Use the `Enable?` switch to specify if the alert is enabled (Yes) or disabled (No).
- Use the controls in the `Interval` box to specify how often the alert should confirm if the alert conditions are satisfied. Use the Minutes selector to specify an interval value. Use the Default switch to set or reset the Minutes value to the default (recommended) value for the selected template.
- Use controls in the `History retention` box to specify the number of days that PEM will store data collected by the alert. Use the Days selector to specify the number of days that the data will be stored. Use the Default switch to set or reset the Days value to the default value (30 days).
- Use controls in the `Threshold values` box to define the triggering criteria for the alert. When the value specified in the Threshold Values fields evaluates to greater-than or less-than the system value (as specified with the Operator), PEM will raise a Low, Medium or High level alert:
- Use the `Operator` drop-down listbox to select the operator that PEM will use when evaluating the current system values:
  - Select a greater-than sign (>) to indicate that the alert should be triggered when the system values are greater than the values entered in the Threshold values fields.
  - Select a less-than sign (<) to indicate that the alert should be triggered when the system values are less than the values entered in the Threshold values fields.
- Use the `threshold` fields to specify the values that PEM will compare to the system values to determine if an alert should be raised. Please note that you must specify values for all three thresholds (Low, Medium, and High):
  - Enter a value that will trigger a low-severity alert in the `Low` field.
  - Enter a value that will trigger a medium-severity alert in the `Medium` field.
  - Enter a value that will trigger a high-severity alert in the `High` field.

The `Parameter Options` table contains a list of parameters that are required by the selected template; the table displays both pre-defined parameters, and parameters for which you must specify a value. Please note that you must specify a value for any parameter that displays a prompt in the Value column.

PEM can send a notification or execute a script if an alert is triggered, or if an alert is cleared. Use the `Notification` tab to specify how PEM will behave if an alert is raised.

The alert editor Notification tab

Use the fields in the `Email notification` box to specify the email group that will receive an email notification if the alert is triggered at the specified level. Use the `Email Groups` tab to create an email group that contains the address of the user or users that will be notified when an alert is triggered. To access the `Email Groups` tab, click the Email Groups icon located in the Quick Links menu of the `Manage Alerts` tab.

- To instruct PEM to send an email when a specific alert level is reached, set the slider next to an alert level to Yes, and use the drop-down listbox to select the pre-defined user or group that will be notified.

Please note that you must configure the PEM Server to use an SMTP server to deliver email before PEM can send email notifications.

Use the `Trap notification` options to configure trap notifications for this alert:

- Set the `Send trap` slider to `Yes` to send SNMP trap notifications when the state of this alert changes.
- Set the `SNMP Ver` to `v1` , `v2` , or `v3` to identify the SNMP version.
- Use the `Low alert` , `Med alert` and `High alert` sliders to select the level(s) of alert that will trigger the trap. For example, if you set the slider next to High alert to Yes, PEM will send a notification when an alert with a high severity level is triggered.

Please note that you must configure the PEM Server to send notifications to an SNMP trap/notification receiver before notifications can be sent. For sending SNMP v3 traps, pemAgent will use 'User Security Model(USM)' which is in charge of authenticating, encrypting, and decrypting SNMP packets.

Also note while sending SNMP v3 traps, agent will create snmp_boot_counter file. This file will get created in location mentioned by batch_script_dir parameter in agent.cfg, if this parameter is not configured or if directory is not accessible due to authentication restrictions then in operating systems temporary directory, if that is also not possible then in user's home directory.

Use the field in the `Nagios notification` box to instruct the PEM server to notify Nagios network-alerting software when the alert is triggered or cleared.

- Set the `Submit passive service check result to Nagios` switch to `Yes` to instruct the PEM server to notify Nagios when the alert is triggered or cleared.

Use the fields in the `Script execution` box to (optionally) define a script that will be executed if an alert is triggered, and to specify details about the script execution.

- Set the `Execute script` slider to `Yes` to instruct PEM to execute the provided script if an alert is triggered.
- Set the `Execute on alert cleared` slider to `Yes` to instruct PEM to execute the provided script when the situation that triggered the alert has been resolved.
- Use the radio buttons next to `Execute script on` to indicate that the script should execute on the PEM Server or the Monitored Server.
- Provide the script that PEM should execute in the `Code` field. You can provide a batch/shell script, or SQL code. Within the script, you can use placeholders for the following:

  `%AlertName%` - this placeholder will be replaced with the name of the triggered alert.

  `%ObjectName%` - this placeholder will be replaced with the name of the server or agent on which the alert was triggered.

  `%ThresholdValue%` - this placeholder will be replaced with the threshold value reached by the metric when the alert triggered.

  `%CurrentValue%` - this placeholder will be replaced with the current value of the metric that triggered the alert.

  `%CurrentState%` - this placeholder will be replaced with the current state of the alert.

  `%OldState%` - this placeholder will be replaced with the previous state of the alert.

`%AlertRaisedTime%` - this placeholder will be replaced with the time that the alert was raised, or the most recent time that the alert state was changed.

To invoke a script on a Linux system, you must modify the entry for *batch_script_user* parameter of agent.cfg file and specify the user that should be used to run the script. You can either specify a non-root user or root for this parameter. If you do not specify a user, or the specified user does not exist, then the script will not be executed. Restart the agent after modifying the file.

To invoke a script on a Windows system, set the registry entry for *AllowBatchJobSteps* to true and restart the PEM agent. PEM registry entries are located in HKEY_LOCAL_MACHINE/Software/Wow6432Node/EnterpriseDB/

When you have defined the alert attributes, click the edit icon to close the alert definition editor, and then the save icon (in the upper-right corner of the `Alerts` table). To discard your changes, click the refresh icon; a popup will ask you to confirm that you wish to discard the changes.

Modifying or Deleting an Alert Use the `Alerts` table to manage an existing alert or create a new alert. Highlight an object in the PEM client tree control to view the alerts that monitor that object.

The Alerts table

You can modify some properties of an alert in the `Alerts` table:

- The `Alert name` column displays the name of the alert; to change the alert name, simply replace the name in the table, and click the save icon.
- The `Alert template` column displays the name of the alert template that specifies properties used by the alert. You can use the drop-down listbox to change the alert template associated with an alert.
- Use the `Alert enable?` switch to specify if an alert is enabled (Yes) or disabled (No).
- Use the `Interval` column to specify how often PEM should check to see if the alert conditions are satisfied. Set the `Default` switch to `No` and specify an alternate value (in Minutes), or return the Default switch to `Yes` to reset the value to its default setting. By default, PEM will check the status of each alert once every minute.
- Use the `History retention` field to specify the number of days that PEM will store data collected by the alert. Set the `Default` switch to `No` and specify an alternate value (in Days), or return the Default switch to `Yes` to reset the value to its default setting. By default, PEM will recommend storing historical data for 30 days.

After modifying an alert, click the save icon (located in the upper-right corner of the table) to make your changes persistent.

Click the edit icon to the left of an alert name to open an editor that provides access to the complete alert definition to modify other alert attributes.

The Alert details dialog

Use fields on the `Alert details` dialog to modify the definition of the selected alert. When you've finished modifying the alert definition, click `Save` to preserve your changes, or `Cancel` to exit the dialog without saving any changes.

Deleting an Alert

To mark an alert for deletion, highlight the alert name in the Alerts table and click the delete icon to the left of the name; the alert will remain in the list, but in red strike-through font.

Deleting an alert

The delete icon acts as a toggle; you can undo the deletion by clicking the delete icon a second time; when you click the Save icon, the alert definition will be permanently deleted.

Copying an Alert To speed up the deployment of alerts in the PEM system, you can copy alert definitions from one object to one or more target objects.

To copy alerts from an object, highlight the object in the PEM client tree control on the main PEM window, and select the `Copy Alerts...` option from the `Management` menu. When the `Manage Alerts` tab opens, click the Copy Alerts icon (located in the Quick Links toolbar) to open the `Copy Alert Configuration` dialog.

The Copy Alert Configuration dialog

The `Copy Alert Configuration` dialog copies all alerts from the object highlighted in the PEM client tree control to the object or objects selected on the dialog. Expand the tree control to select a node or nodes to specify the target object(s). The tree control displays a red warning indicator next to the source object.

To copy alerts to multiple objects at once, select a parent node of the target(s). For example, to copy the alerts from one table to all tables in a schema, you can simply select the checkbox next to the schema. PEM will only copy alerts to targets that are of the same type as the source object.

Check the `Ignore duplicates` radio button to prevent PEM from updating any existing alerts on the target objects with the same name as those being copied. Use the `Replace duplicates` option to replace existing alerts with alerts of the same name from the source object.

Click the `Configure Alerts` button to proceed to copy the alerts from the source object to all objects of the same type in, or under those objects selected on the `Copy Alert Configuration` dialog.

Audit Log Alerting PEM provides alert templates that allow you to use the `Alerting` dialog to create an alert that will trigger when an `ERROR` or `WARNING` statement is written to a log file for a specific server or agent. To open the `Alerting` dialog, highlight the name of the server or agent in the PEM client Object browser tree control, and select `Alerting...` from the `Management` menu.

To create an alert that will notify you of ERROR or WARNING messages in the log file for a specific server, create an alert that uses one of the following alert templates:

> Number of ERRORS in the logfile on server M in last X hours
>
> Number of WARNINGS in the logfile on server M in last X hours
>
> Number of ERRORS or WARNINGS in the logfile on server M in last X hours

To create an alert that will notify you of ERROR or WARNING messages for a specific agent, create an alert that uses one of the following alert templates:

> Number of ERRORS in the logfile on agent M in last X hours
>
> Number of WARNINGS in the logfile on agent M in last X hours
>
> Number of ERRORS or WARNINGS in the logfile on agent M in last X hours

Please note that this functionality is supported only on Advanced Server.

Creating an Email Group Postgres Enterprise Manager monitors your system for conditions that require user attention. You can use an email group to specify the email addresses of users that the server will notify if current values deviate from threshold values specified in an alert definition. An email group has the flexibility to notify multiple users, or target specific users during user-defined time periods.

Please note that you must configure the PEM Server to use an SMTP server to deliver email before PEM can send email notifications.

Use the `Email Groups` tab to configure groups of SMTP email recipients. To access the `Email Groups` tab, select `Manage Alerts...` from the PEM client's `Management` menu; when the `Manage Alerts` tab opens, select `Email Groups` from the Quick Links toolbar.

The Email Groups tab

The `Email Groups` tab displays a list of the currently defined email groups. Highlight a group name and click the Edit icon (at the far left end of the row) to modify an existing group.

To define a new email group, click the Add icon (+) in the upper-right corner of the `Email Groups` table. The `Email Group` definition dialog opens.

Adding an email group

Use the `Email Group` dialog to define an email group and its members:

- Provide a name for the email group in the `Group Name` field.

Each row within the email group definition will associate a unique set of email addresses with a specific time period. When an alert is triggered, the server will evaluate the times specified in each row and send the message to those group members whose definitions are associated with the time that the alert triggered.

Click the Add icon (+) in the group members table to open the `Options` tab, and add the member addresses that will receive notifications for the time period specified:

- Enter a comma-delimited list of recipient addresses in the `Reply` to Addresses field.
- Enter a comma-delimited list of addresses that will receive a copy of the email in the `CC Addresses` field.
- Enter a comma-delimited list of addresses that will receive a copy of the email (without the knowledge of other recipients) in the `Bcc Addresses` field.
- Enter the email address that messages to this group should be sent from in the `From Address` field.
- Use the `Subject prefix` field to provide a message that will be added to the start of each subject line when a notification is sent.
- Use the `From Time` and `To Time` time selectors to specify the time range for notifications to the group member(s) that are identified on this row. Provide the From Time and To Time values in the locale of the PEM client host, and the PEM server will translate the time into other time zones as required.

When you've identified the member or members that will receive an email during a specific time period, click the Add icon to add a row to the table, and specify another time period and the email addresses that will be notified during those hours. When you've finished defining the email group, click the Save icon.

To delete an email group, highlight the name of the group in the `Email Group` table and click the Delete icon (located to the left of the group name).

Deleting an email group

The group name will be displayed in the `Email Group` table in red; click the `Save` icon to make the change persistent and remove the group from the table.

After creating the email group, you can use the `Manage Alerts` tab to set up the `Notification` details for an alert that will direct notifications to the group.

Using PEM with Nagios

Using PEM with Nagios

The PEM server can send a passive alert result to Nagios network-alerting software when a user-defined alert is triggered. To instruct the PEM server to notify Nagios of a triggered alert, you must:

- Enable Nagios notification for each alert that will trigger a notification from the PEM server to Nagios. Please note that PEM alerting must be configured before you create the `host.cfg` file, the `services.cfg` file, or configure Nagios.
- Configure Nagios-related behaviors of the PEM server.
- Create the `host.cfg` and `services.cfg` configuration files.
- If necessary, modify the Nagios configuration file and restart the server.

After configuring the server to enable Nagios alerting, any triggered alerts will send a passive check result to the Nagios service. The syntax of a passive alert is:

```
<timestamp> PROCESS_SERVICE_CHECK_RESULT; <host_name> ; <service_name> ; <service_status>
```

Where:

`timestamp` is the date and time that the alert was triggered.

`host_name` is the name of the server or agent.

`service_name` is the name of the alert.

`service_status` is the numeric service status value:

0 if the service status is OK
1 if the service status is WARNING
2 if the service status is CRITICAL
3 if the service status is UNKNOWN

The PEM server uses the following rules to evaluate the service status:

- If the PEM alert level is CLEARED, the warning message will read OK.
- If the PEM alert level is LOW, the warning message will read WARNING.
- If the `is_nagios_medium_alert_as_critical` flag (specified in the PEM server configuration dialog) is set to FALSE and the alert level MEDIUM, the warning message will read WARNING.
- If the `is_nagios_medium_alert_as_critical` flag (specified in the PEM server configuration dialog) is set to TRUE and the alert level is MEDIUM, the warning message will read CRITICAL.
- If the PEM alert level is `HIGH` , the warning message will read `CRITICAL` .

Enabling Nagios Notification for an Alert   Enabling Nagios Notification for an Alert

The PEM server maintains a unique set of notification properties for each enabled alert. Use the `Notification` tab of the `Manage Alerts` tab to specify that (when triggered), a given alert will send an alert notice to Nagios.

To modify the notification properties of an alert, right-click on the name of the object monitored by the alert, and select `Manage Alerts...` from the `Management` menu. When the `Manage Alerts` tab opens, locate the alert, and then click the edit button to the left of the alert name in the `Alerts` list. When the edit pane opens, select the `Notification` tab.

The Notification tabg

To enable Nagios notification, move the slider next to `Submit passive service check result` to Nagios to `Yes` ; before exiting the `Manage Alerts` tab, click the save icon to preserve your changes.

Configuring Nagios-related behavior of the PEM Server   Configuring Nagios-related behavior of the PEM Server

You can use the `Server Configuration` dialog to provide information about your Nagios configuration to the PEM server. To open `Server Configuration` dialog, select `Server Configuration...` from the PEM client's `Management` menu.

Specify Nagios properties in the Server Configuration dialog

Four server configuration parameters specify information about your Nagios installation and PEM server behavior related to Nagios:

- Use the `nagios_cmd_file_name` parameter to specify the location of the Nagios pipeline file that will receive passive check alerts from PEM. The default value of this parameter is `/usr/local/nagios/var/rw/nagios` . If your `nagios.cmd` file resides in an alternate location, specify the file location in the Value field.
- Move the slider in the `nagios_enabled` parameter to `Yes` to instruct the PEM server to send passive check alerts to Nagios.
- Use the `nagios_medium_alert_as_critical` slider to specify the warning severity that the PEM server will pass to Nagios if a medium alert is triggered:

  If the `is_nagios_medium_alert_as_critical` flag is set to FALSE and the alert level is MEDIUM, the warning message will read WARNING.

  If the `is_nagios_medium_alert_as_critical` flag is set to TRUE and the alert level is MEDIUM, the warning message will read CRITICAL.

- Use the `nagios_spool_retention_time` parameter to specify the number of days of notification history that will be stored on the PEM server. The default value is 7 days.

After modifying parameter values, click the save icon (in the upper-right corner of the `Server Configuration` dialog) to preserve your changes.

Creating the hosts.cfg and services.cfg File The `templates.cfg` file (by default, located in `/usr/local/nagios/etc/objects`) specifies the properties of a generic-host and generic-service. The properties specify the parameters used in the `hosts.cfg` and `services.cfg` files.

In most cases (when PEM is installed in a default configuration), you will not be required to modify the `templates.cfg` file before creating the `hosts.cfg` and `services.cfg` files. If necessary, you can modify the `templates.cfg` file to specify alternate values for parameters or to create new templates.

Before modifying the Nagios configuration file, use the following command to create a `hosts.cfg` file that contains information about the PEM hosts that reside on the local system:

```
psql -U postgres -p 5433 -d pem -A -t -c "select pem.create_nagios_host_config('generic-h
```

Then, use the following command to create a `services.cfg` file that contains information about the PEM services that reside on the local system:

```
psql -U postgres -p 5433 -d pem -A -t -c "select pem.create_nagios_service_config('generi
```

If you wish to use a `custom template.cfg` file entry, specify the entry name in place of generic-host or generic-service in the above commands.

Modifying the Nagios Configuration File Modifying the Nagios Configuration File

After creating the `host.cfg` and `services.cfg` files, you must specify their location in the Nagios configuration file (by default, `/usr/local/nagios/etc/nagios.cfg`). Modify the configuration file, adding entries that specify the location of the files:

```
cfg_file=/usr/local/etc/objects/hosts.cfg
```

```
cfg_file=/usr/local/etc/objects/services.cfg
```

You can use the following command to confirm that Nagios is properly configured:

```
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

After confirming that Nagios is configured correctly, restart the Nagios service:

```
/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
```

7.5 Capacity Manager

Capacity Manager

PEM's Capacity Manager analyzes collected statistics (metrics) to generate a graph or table that displays the historical usage statistics of an object, and can project the anticipated usage statistics for an object. You can configure Capacity Manager to collect and analyze metrics for a specific host, server, database, or database object.

You can tailor the content of the Capacity Manager report by choosing a specific metric (or metrics) to include in the report, the time range over which the metrics were gathered, and a high or low threshold for the metrics analyzed. You can also specify a start and end date for the Capacity Manager report. If the end date of the report specifies a time in the future, Capacity Manager will analyze the historical usage of the selected object to extrapolate the projected object usage in the future.

To open Capacity Manager, select the `Capacity Manager...` option from the PEM client `Management` menu; the Capacity Manager wizard opens, displaying a tree control on the `Metrics` tab.

The Capacity Manager dialog

Expand the tree control on the `Metrics` tab to review the metrics for the node that you wish to analyze. Check the box to the left of the name of the metric to include the metric in your report.

The Metrics tree control

Capacity Manager will use the aggregation method specified with the Aggregation drop-down listbox (located at the bottom of the Metrics tab). The aggregation method instructs Capacity Manager how to evaluate and plot the metric values. Select from:

- `Average` : Use the average of the values recorded during the time period.
- `Maximum` : Use the maximum value recorded during the time period.
- `Minimum` : Use the minimum value recorded during the time period.
- `First` : Use the first value recorded during the time period.

To remove a metric from the Capacity Manager report, uncheck the box to the left of the name of a metric.

Move the slider next to `Graph/chart metrics individually?` to `Yes` to instruct Capacity Manager to produce a separate report for each metric selected on the Metrics tab. If the option is set to No, all selected metrics will be merged into a single graph or table.

Click the `Generate` button to display the report onscreen (accepting the default configuration options), or use the Options tab to customize sampling boundaries, report type and report destination. Please note that the times displayed on the Options tab are the time zone in which the PEM client resides.

Specify the time period, type, and destination of the report

Use the fields within the `Time Period` box to define the boundaries of the Capacity Manager report:

- Use the `Period` drop-down listbox to select the type of time period you wish to use for the report. You can select:

Start time and end time	Specify a start date and an end date/time for the report.
Start time and threshold	Specify a start date and time, and a threshold to determine the end time and date
Historical days and extrapolated days	Specify a start date for the report that is a number of days in the past, and an end
Historical days and threshold	Specify a start date that is a number of days in the past, and end it when a thresho

After specifying the type of time period for the report, select from other options in the Time Period box to define the time period for the report:

- Use the date and time selectors next to the `Start time` field to specify the starting date and time of the sampling period, or select the number of Historical day(s) of data to include in the report. The date and time specified in the Start time field must not be later than the current date/time.

  By default, Capacity Manager will select a start time that is one week prior to the current date and time.

- The end boundary for the report can be a time, a number of days in the future, or the point at which a selected metric reaches a user-specified threshold value. Use the date and time selectors next to the `End time` field to specify an end boundary for the report, or select the number of Extrapolated day(s) of data to include in the report. The time specified in the End time field must be later than the time specified in the Start time field.

  Note that if you select an end date and time in the future, Capacity Manager will use historical usage information to extrapolate anticipated future usage. Since the projected usage is based on the sampling of historical data, the accuracy of the future usage trend will improve with a longer sampling period.

  To specify a threshold value, use the drop-down listbox in the Threshold field to select a metric, an operator (Exceeds or Falls below), and enter a target value for the metric. If you choose to define the end of the report using a threshold, the Capacity Manager report will terminate when the value for the selected metric exceeds or falls below the specified value.

The `cm_max_end_date_in_years` configuration parameter defines a default time value for the end boundary of a Capacity Manager report. If you specify a threshold value as the end boundary of a report, and the anticipated usage of the boundary is not met before the maximum time has passed, the report will terminate at the time specified by the cm_max_date_in_years parameter. By default, cm_max_end_date_in_years is 5; you can use the Server Configuration dialog to modify the value of cm_max_end_date_in_years.

The fields in the `Report` box specify the report type and destination. Use the `Include on report` radio buttons to specify the type of report produced by Capacity Manager:

- Select `Graph` to instruct Capacity Manager to display the report in the form of a line graph in the PEM client window.
- Select `Table of data` to instruct Capacity Manager to display a table containing the report data in the PEM client window.
- Select `Graph and table of data` to instruct Capacity Manager to display both a line graph and a data table in the PEM client window.

Use the `Report destination` radio buttons to instruct Capacity Manager where to display or save the report:

- Select `New` tab to instruct Capacity Manager to display the report on a new tab in the PEM client. You must select New tab to display the first generation of a Capacity Manager report; for subsequent reports, you may select Previous tab.
- Select `Previous` tab to instruct Capacity Manager to re-use a previously opened tab when displaying the report.
- Select `Download` the report as a file and specify a file name to instruct Capacity Manager to write the report to the specified file.

When you have specified the report boundaries and selected the type and destination of the Capacity Manager report, click the `Generate` button to create the report.

The Capacity Manager report

Reports saved to file are stored in HTML format. You can review a Capacity Manager report with any web browser that supports Scalable Vector Graphics (SVG). Browsers that do not support SVG will be unable to display a Capacity Manager graph and may include unwanted characters.

Capacity Manager Templates

After defining a report, you can save the definition as a template for future reports. Capacity Manager report templates may be accessed by all PEM users. To save a report definition as a template:

1. Use the `Metrics and Options` tabs to define your report.
2. Click the `Save` button to open the `Save Template` dialog.

Saving a Capacity Manager Template

3. Provide a report name in the `Title` field, select a location to store the template in the tree control.
4. Click `OK` .

When creating a report, you can use the `Load Template` button to browse and open an existing template. Once opened, the report definition may be modified if required, and optionally saved again, either as a new template, or overwriting the original template.

Use the `Manage Templates` button open a dialog that allows you to rename or remove unwanted templates.

7.6 Audit Manager

Audit Manager

You can use the PEM Audit Manager to simplify audit log configuration for Advanced Server instances. With the Audit Manager, you can configure logging attributes such as:

- How often log files are to be collected by PEM

- The type of database activities that are included in the log files
- How often (and when) log files are to be rotated

Audit logs may include the following activities:

- All connections made to the database instance
- Failed connection attempts
- Disconnections from the database instance
- All queries (SELECT statements)
- All DML statements (INSERT, UPDATE, DELETE)
- All DDL statements (e.g., CREATE, DROP, ALTER)

Once the audit logs are stored on the PEM server, you can use the Audit Log dashboard to review the information in an easy-to-read form. The Audit Log dashboard allows you to filter the log file by timestamp range (when an activity occurred), the database on which the activity occurred, the user performing the activity, or the type of command being invoked.

Setting the Advanced Server Instance Service ID

To configure logging for an Advanced Server instance, the server must be a PEM-managed server with a bound agent, and the server registration must include the name of a service script. When registering a new server, include the service name in the Service ID field on the Advanced tab of the New Server dialog.

Before adding a service name to an existing (registered and connected) server, you must disconnect the server. Right click on the server name, and select `Disconnect server` from the context menu. Then, right click on the server name and select `Properties` from the context menu. Select the `Advanced` tab, and add a service name to the `Service ID` field.

The Service ID of the Advanced Server instance

The Service ID field allows the PEM server to stop and start the service.

- The name of the Advanced Server 11 service script is `edb-as-11` .
- The name of the Advanced Server 10 service script is `edb-as-10` .
- The name of the Advanced Server 9.6 service script is `edb-as-9.6` .
- The name of the Advanced Server 9.5 (or prior) service script is ppas-9.*x*, where *x* specifies the version.
- The name of the PostgreSQL 9.6 service script is `postgresql-11` .
- The name of the PostgreSQL 9.6 service script is `postgresql-10` .
- The name of the PostgreSQL 9.6 service script is `postgresql-9.6` .

Setting the EDB Audit Configuration Probe

Before configuring audit logging of Advanced Server servers, you must ensure that the EDB Audit Configuration probe is enabled. To open the `Manage Probes` tab and check the status of the probe, right click on the name of a registered Advanced Server server in the tree control, and select `Manage Probes...` from the `Management` menu.

Ensure that the `Enabled` column in the `Probe Configuration` dialog is set to `Yes` for the `EDB Audit Configuration probe` .

The EDB Audit Configuration probe

If EDB Audit Configuration is not enabled, use the `Enabled?` switch on the Manage Probes tab to enable it.

Configuring Audit Logging with the Audit Manager

To open the `Audit manager` wizard, select `Audit Manager...` from the `Management` menu. The `Audit manager - Welcome` dialog opens.

The Audit Manager Welcome dialog

Click `Next` to continue.

Select the servers you wish to configure for auditing

Use the Select servers tree control to specify the servers to which the auditing configuration will be applied. To make a server available in the tree control, you must provide the `Service ID` on the `Advanced tab` of the `Create - Server` dialog when registering a server for monitoring by PEM. Note that only EDB Postgres Advanced Server supports auditing; PostgreSQL servers will not be included in the tree control.

Click `Next` to continue.

The `Auditing Parameters Configuration` dialog lets you enable or disable auditing and choose how often log records are collected into PEM.

The Auditing Parameters Configuration dialog

Use the fields on the `Auditing parameters configuration` dialog to specify auditing preferences:

- Use the `Auditing` switch to Enable or Disable auditing on the specified servers.
- Use the `Audit destination` drop-down to select a destination for the audit logs; select File or Syslog. Please note this feature is supported on Advanced Server 10 and newer releases only.
- Use the `Import logs to PEM` switch to instruct PEM to periodically import log records from each server to the PEM Server. Set the switch to Yes to import log files; the default is No.
- Use the `Collection frequency` drop-down listbox to specify how often PEM will collect log records from monitored servers when log collection is enabled.
- Use the `Log format` drop-down listbox to select the raw log format that will be written on each server. If log collection is enabled, the PEM server will use CSV format.
- Use the `File name` field to specify the format used when generating log file names. By default, the format is set to `audit-%Y-%m-%d_%H%M%S` where:

  `audit` is the file name specified in the Audit Directory Name field
  `Y` is the year that the log was stored
  `m` is the month that the log was stored
  `d` is the day that the log was stored
  `H` is the hour that the log was stored
  `M` is the minute that the log was stored
  `S` is the second that the log was stored

- Check the box next to `Change Log Directory for selected servers?` and use the `Audit Directory Name` field to specify a directory name to which the audit logs will be written. The directory will reside beneath the data directory on the PEM server.

Use fields in the `Log directory` box to specify information about the directory in which the log files will be saved:

- Move the `Change log directory for selected servers?` switch to Yes to enable the Directory name field.
- Use the `Directory name` field to specify the name of the directory on each server into which audit logs will be written. The directory specified will be created as a sub-directory of the data directory on the server.

Click `Next` to continue.

The `Audit log configuration` dialog is only available if you have enabled auditing on the Auditing parameters configuration dialog.

The Audit Log Configuration dialog

Use the controls on the `Audit log configuration` dialog to specify log configuration details that will be applied to each server:

- Use the `Connection attempts` switch to specify if connection attempts should be logged:

  `None` to disable connection logging.

  `All` to indicate that all connection attempts will be logged.

  `Failed` to log any connection attempts that fail.

- Use the `Disconnection attempts` switch to specify if disconnections should be logged. Specify:

  `None` to specify that disconnections should not be logged.

  `All` to enable disconnection logging.

- Use the `Log statements` field to specify the statement types that will be logged. Click within the field, and select from:

  `Select` - All statements that include the `SELECT` keyword will be logged.

  `Error` - All statements that result in an error will be logged.

  `DML` - All DML (Data Modification Language) statements will be logged.

  `DDL` - All DDL (Data Definition Language) statements (those that add, delete or alter data) will be logged.

  Check the box next to `Select All` to select all statement types.

  Check the box next to `Unselect All` to deselect all statement types.

- Use the `Audit tag` field to specify a tracking tag for the collected logs. Please note that audit tagging functionality is available only for Advanced Server versions 9.5 and later. If you are defining auditing functionality for multiple servers, and one or more of the servers are version 9.5 or later, this field will be enabled, but if selected, tagging functionality will only apply to those servers that are version 9.5 or later.

Use the fields in the `Log rotation` box to specify how the log files are managed on each server:

- Use the `Enable?` switch to specify that logfiles should be rotated. Please note that a new log file should be used periodically to prevent a single file becoming unmanageably large.
- Use the `Day drop-down` listbox to select a day or days on which the log file will be rotated.
- Use the `Size (MB)` field to specify a size in megabytes at which the log file will be rotated.
- Use the `Time (seconds)` field to specify the number of seconds between log file rotations.

Click `Next` to continue:

The Schedule Auditing Changes dialog

Use the `Schedule Auditing Changes` dialog to determine when auditing configuration changes are to take effect.

- Select `Configure logging now?` if you want the auditing configuration changes to take place immediately. The affected database servers will be restarted so the auditing changes can take effect.
- Use the `Time?` selector to schedule the auditing configuration changes to take place at some point in the future. Select the desired date and time from the drop-down lists. The affected database servers will be restarted at the specified date/time to put the auditing changes into effect.

Click `Finish` to complete the auditing configuration process.

The Audit Manager will schedule a job to apply the configuration to each server. The job will consist of two tasks: one to update the audit logging configuration on the server, and one to restart the server with the new configuration.

You can use the `Scheduled Tasks` tab to review a list of Scheduled jobs. To open the `Scheduled Tasks` tab, highlight the name of a server or agent and select `Scheduled Tasks...` from the `Management` menu.

Viewing the Log with the Audit Log Dashboard

Use the Audit Log dashboard to view the audit log from Advanced Server database instances.

To open the `Audit Log` dashboard, right click on a server or agent node, and select `Audit Log Analysis` from the `Dashboards` menu. You can also open the Audit Log dashboard by navigating through the `Dashboards` menu (located on the `Management` menu).

The Audit Log dashboard

The Audit Log dashboard displays the audit records in reverse chronological order (newest records at the top, oldest records towards the bottom).

To view older audit records that do not appear in the window, use the vertical scroll bar controlling the list of audit records (the innermost scroll bar of the two located on the right-hand side of the window). As you move the scroll bar towards the bottom of the window, older audit records are continuously loaded and displayed.

You can use filtering to limit the number of audit records that are displayed. Click `Show Filters` to expose the filters panel.

The Audit Log dashboard filters panel

Use the fields in the `filters panel` to provide certain selection criteria for the audit records you wish to display.

- Use the `Start` field to specify a start date for the report. Click the mouse button in the field to open a calendar and select a start date.
- Use the `End` field to specify an end date for the report. Click the mouse button in the field to open a calendar and select an end date.
- Use the `User` field to display only those entries where the activity was initiated by the given Postgres user.
- Use the `Database` field to display only those entries where the activity was issued on the given database.
- Use the `Command type` field to display only those entries where the activity was of the given type. Command types you can specify are idle, authentication, and SELECT. (For viewing SQL statements from user applications, specify the idle command type.)

Click `Filter` to apply the filtering criteria to the log entries.

7.7 Log Manager

Log Manager

You can use the PEM Log Manager to simplify server log configuration for Postgres instances. With the Log Manager, you can modify all of your server log parameters with a click:

- Where log files are written
- How often log files are written
- The type of information written to log files
- The format of log file entries
- Log rotation properties

To configure logging for a Postgres instance, the server must be registered as a PEM-managed server, and the registration information must include the name of a service script.

To open the `Log Manager`, select the `Log Manager...` option from the `Management` menu of the PEM client. The wizard opens, welcoming you to the Log Manager.

The Log Manager welcome dialog

Click `Next` to continue to the `Server selection` dialog.

The Log Manager Server selection dialog

The `Server selection` dialog displays a list of the server connections monitored by PEM. Check the box next to the name of a server (or servers) to which the Log Manager wizard will apply the specified configuration.

Log Manager is disabled for any server displaying a red exclamation mark to the left of its name in the Server selection tree control; there are several reasons that a server may not be enabled:

- Only a server that specifies a `Service ID` on the `Advanced` tab of the `Properties` dialog can be configured by Log Manager.

  To provide a service ID, right click on the server name in the tree control, and select `Disconnect Server` from the context menu; if prompted, provide a password. Then, open the context menu for the server, and select `Properties`. Navigate to the `Advanced` tab, and provide the name of the service in the `Service ID` field; click `Save` to save your change and exit the dialog.
- If the PEM agent bound to the server does not have sufficient privileges to restart the server, the server will be disabled.
- If the PEM agent bound to the server is an older version than the associated PEM server, the server will be disabled.

Click `Next` to continue.

The Log Manager Log configuration dialog

Use the options on the `Log configuration` dialog to specify how often log files will be inported to PEM and to specify log rotation details:

Options within the `Import Logs` box specify how often log files will be imported to PEM:

- Use the switch next to the `Import logs to PEM` label to specify if log files will be imported to PEM and displayed on the Server Log Analysis dashboard.
- Use the `Import Frequency` drop-down list box to specify how often log files are imported to PEM.

Use the fields in the `Log rotation configuration` box to specify the maximum length (lifespan or size) of a log file:

- Use the `Rotation Size` field to specify the maximum size in megabytes of an individual log file. The default value is 10 MB; when set to 0, no limit is placed on the maximum size of a log file.
- Use the `Rotation Time` field to specify the number of whole days that should be stored in each log file. The default value is 1 day.

Use the `Truncation on Rotation` switch to specify server behavior for time-based log file rotation:

- Select ON to specify that the server should overwrite any existing log file that has the same name that a new file would take.
- Select OFF to specify that the server should append any new log file entries to an existing log file with the same name that a new log file would take. This is the default behavior.

Click `Next` to continue.

The Where to Log dialog

Use the fields on the `Where to` log dialog to specify where log files should be written.

- Select an option from the `Log Destination` box to specify a destination for the server log output:
  - Set the `stderr` switch to Yes to specify that log files should be written to stderr.
  - Set the `csvlog` switch to Yes to specify that log files should be written to file in a comma-separated value format. This option is automatically enabled (and no longer editable) if you have selected Import logs to PEM on the Schedule dialog; if you are not importing server log files to PEM, this option is editable.
  - Set the `syslog` switch to Yes to specify that log files should be written to the system log files.
  - On Windows, set the `eventlog` switch to Yes to specify that log files should be written to the event log.
- Use the options within the `Log collection` box to specify your collection preferences:
  - Set the `Log Collector` switch to Enable to instruct the server to re-direct captured log messages (directed to STDERR) into log files.

  - Set the `Log Silent Mode` switch to Enable to instruct the server to run silently in the background, disassociated from the controlling terminal.
- Use options in the `Log Directory` box to specify log file location preferences:
  - Set the `Change log directory for selected servers?` switch to Yes to specify that each set of log files should be maintained in a separate directory.
  - Use the `Directory name` field to specify the directory to which log files will be written. The directory will reside beneath the pg_log directory under the installation directory of the monitored server.
- Use the `Log File Name` field to specify a format for the log file name. If set to `DEFAULT`, the format is `enterprisedb-%Y-%m-%d_%H%M%S`, where:
  - `enterprisedb` is the file name prefix
  - `Y` is the year that the log was stored
  - `m` is the month that the log was store
  - `d` is the day that the log was stored
  - `H` is the hour that the log was stored
  - `M` is the minute that the log was store
  - `S` is the second that the log was stored

When logging to syslog is enabled:

- Use the `Syslog Facility` drop-down list box to specify which syslog facility should be used.
- Use the `Syslog Ident` field to specify the program name that will identify Advanced Server entries in system logs.

Click `Next` to continue.

The Log Manager When to Log dialog

Use the fields on the `When to log` dialog to specify which events will initiate a log file entry. The severity levels (in order of severity, from most severe to least severe) are:

- `panic` - Errors that cause all database sessions to abort.
- `fatal` - Errors that cause a session to abort.
- `log` - Information messages of interest to administrators.
- `error` - Errors that cause a command to abort.
- `warning` - Error conditions in which a command will complete but may not perform as expected.
- `notice` - Items of interest to users. This is the default.
- `info` - Information implicitly requested by the user.
- `debug5` through `debug1` - Detailed debugging information useful to developers.
- Use the `Client min messages` drop-down list box to specify the lowest severity level of message sent to the client application.
- Use the `Log min messages` drop-down list box to specify the lowest severity level that will be written to the server log.
- By default, when an error message is written to the server log, the text of the SQL statement that initiated the log entry is not included. Use the `Log min error statement` drop-down list box to specify a severity level that will trigger SQL statement logging. If a message is of the specified severity or higher, the SQL statement that produced the message will be written to the server log.
- Use the `Log min duration statement` drop-down list box to specify a statement duration (in milliseconds); any statements that exceed the specified number of milliseconds will be written to the server log. A value of -1 disables all duration-based logging; a value of 0 logs all statements and their duration.
- Use the `Log temp files` field to specify a file size in kilobytes; when a temporary file reaches the specified size, it will be logged. A value of -1 (the default) disables this functionality.
- Use the `Log autoVacuum min duration` field to specify a time length in milliseconds; if autovacuuming exceeds the length of time specified, the activity will be logged. A value of -1 (the default) disables this functionality.

Click `Next` to continue.

The Log Manager What to Log dialog

Use the fields on the `What to` log dialog to specify log entry options that are useful for debugging and auditing.

The switches in the `Debug options` box instruct the server to include information in the log files related to query execution that may be of interest to a developer:

- Set the `Parse tree` switch to Yes to instruct the server to include the parse tree in the log file.
- Set the `Rewriter output` switch to Yes to instruct the server to include query rewriter output in the log file.
- Set the `Execution plan` switch to Yes to instruct the server to include the execution plan for each executed query in the log file.

When the `Indent Debug Options Output in Log` switch is set to `Yes` , the server indents each line that contains a parse tree entry, a query rewriter entry or query execution plan entry. While indentation makes the resulting log file more readable, it does result in a longer log file.

Use the switches in the `General Options` box to instruct the server to include auditing information in the log file:

- Set the `Checkpoints` switch to Yes to include checkpoints and restartpoints in the server log.
- Set the `Connections` switch to Yes to include each attempted connection to the server (as well as successfully authenticated connections) in the server log.
- Set the `Disconnections` switch to Yes to include a server log entry for each terminated session that provides the session information and session duration.
- Set the `Duration` switch to Yes to include the amount of time required to execute each logged statement in the server log.
- Set the `Hostname` switch to Yes to include both the IP address and host name in each server log entry (by default, only the IP address is logged). Please note that this may cause a performance penalty.
- Set the `Lock Waits` switch to Yes to instruct the server to write a log entry for any session that waits longer than the time specified in the deadlock_timeout parameter to acquire a lock. This is useful when trying to determine if lock waits are the cause of poor performance.

Use the `Error verbosity` drop-down list box to specify the detail written to each entry in the server log:

- Select default to include the error message, DETAIL, HINT, QUERY and CONTEXT in each server log entry.
- Select terse to log only the error message.
- Select verbose to include the error message, the DETAIL, HINT, QUERY and CONTEXT error information, SQLSTATE error code and source code file name, the function name, and the line number that generated the error.

Use the `Prefix string` field to specify a printf-style string that is written at the beginning of each log file entry. For information about the options supported, please see the log_line_prefix documentation (in the Postgres core documentation), available at:

http://www.postgresql.org/docs/current/static/runtime-config-logging.html

Use the `Statements` drop-down list box to specify which SQL statements will be included in the server log. The default is none; valid options are:

- Specify none to disable logging of SQL statements.
- Specify ddl to instruct the server to log ddl (data definition language) statements, such as CREATE, ALTER, and DROP.
- Specify mod to instruct the server to log all ddl statements, as well as all dml (data modification language) statements, such as INSERT, UPDATE, DELETE, TRUNCATE and COPY FROM.
- Specify all to instruct the server to log all SQL statements.

Click `Next` to continue.

The Schedule Logging Changes dialog

Use options on the `Schedule logging changes` dialog to specify when logging configuration changes will be applied:

- Set the `Configure logging now` switch to `Yes` to specify that your configuration preferences will be enabled, and the server will restart when you have completed the Log Manager wizard.
- Set `Configure logging now` to `No` to use the Schedule it for some other time calendar selector to specify a convenient time for logging configuration preferences to be applied, and the server to restart.

Note that when you apply the configuration changes specified by the Log Manager wizard, the server restart will temporarily interrupting use of the database server for users.

Click `Finish` to exit the wizard, and either restart the server, or schedule the server restart for the time specified on the scheduling dialog.

Reviewing the Server Log Analysis Dashboard

After invoking the Log Manager wizard, and importing your log files to PEM, you can use the `Server Log Analysis` dashboard to review the log files for a selected server. To open the `Server Log Analysis` dashboard, right-click on the name of a monitored server in the PEM client tree control, and navigate through the `Dashboards` menu, selecting `Server Log Analysis` .

The Server Log Analysis dashboard

The header information on the `Server Log Analysis` dashboard displays the date and time that the server was started, the date and time that the page was last updated, and the current number of triggered alerts.

Entries in the `Server Log` table are displayed in chronological order, with the most-recent log entries first. Use the scroll bars to navigate through the log entries, or to view columns that are off of the display.

Headings at the top of the server log table identify the information stored in each column; hover over a column heading to view a tooltip that contains a description of the content of each column.

You can use filtering to limit the number of server log records that are displayed. Click `Show Filters` to expose the filters panel and define a filter.

Defining a Server Log filter

Use the fields within the `filter definition` box to describe the selection criteria that PEM will use to select a subset of a report for display:

- Use the `From` field to specify a starting date for the displayed server log.
- Use the `To` field to specify an ending date for the displayed server log.
- Enter a role name in the `Username` field display only transactions performed by that user.
- Enter a database name in the `Database field` to specify that the server should limit the displayed records to only those transactions that were performed against the specified database.
- Use the `Command Type` field to specify a selection criteria for the commands that will be displayed in the filtered report.

When you've described the criteria by which you wish to filter the server logs, click `Filter` to display the filtered server log in the `Server Log` table.

---

title: "Postgres Log Analysis Expert" 7.7 Postgres Log Analysis Expert

The PEM Log Analysis Expert analyzes the log files of servers that are registered with Postgres Enterprise Manager, and produces a report that provides an analysis of your Postgres cluster's usage based on log file entries. You can use information on the Log Analysis Expert reports to make decisions about optimizing your cluster usage and configuration to improve performance.

Before using the PEM Log Analysis Expert, you must specify the Service ID on the Advanced tab of the Server Properties dialog, and use the Log Manager wizard to enable log collection by the PEM server.

To open the `Postgres Log Analysis Expert` wizard, select the `Postgres Log Analysis Expert...` option from the `Management` menu of the PEM client. The wizard's `Welcome` dialog opens; click `Next` to continue:

The Log Analysis Expert Welcome dialog

The wizard's `Analyzer selection` dialog displays a list of Analyzers from which you can select. Each Analyzer generates a corresponding table, chart, or graph that contains information gleaned from the log files.

The Analyzer selection dialog

Check the box to the left of an Analyzer to indicate that the Log Analysis Expert should prepare the corresponding table, chart or graph. After making your selections, click `Next` to continue to the Server selection tree control.

The Server selection dialog

Use the tree control to specify which servers you would like the Postgres Log Analysis Expert to analyze. If you select multiple servers, the resulting report will contain the corresponding result set for each server in a separate (but continuous) list. Click `Next` to continue to the Report options dialog.

The Report options dialog

Use the fields in the `Options` section to specify the analysis method and the maximum length of any resulting tables:

- Use the `Aggregate method` drop-down to select the method used by the Log Analysis Expert to consolidate data for the selected time span. You can select from:
  - `SUM` instructs the analyzer to calculate a value that is the sum of the collected values for the specified time span.
  - `AVG` instructs the analyzer to calculate a value that is the average of the collected values for the specified time span.
  - `MAX` instructs the analyzer to use the maximum value that occurs within a specified time span.
  - `MIN` instructs the analyzer to use the minimum value that occurs within a specified time span.
- Use the `Time span` field to specify the number of minutes that the analyzer will incorporate into each calculation for a point on a graph. For example, if the Time span is 5 minutes, and the Aggregate method is AVG, each point on the given graph will contain the average value of the activity that occurred within a five minute time span.
- Use the `Rows limit` field to specify the maximum number of rows to include in a table.

Use the fields in the `Time Intervals` section to specify the time range that the Log Analysis Expert will analyze:

- Set `Relative days` to Yes to enable the (+/-)From date field and specify the number of days before or after the date and time selected in the From field.
- Use the `From` field to specify the starting date and time for the analysis.
- Use the `To` field to specify the ending date and time for the analysis.
- Use the `(+/-)` From date selector to specify the number of days before or after the From date that should be included in the analysis.

When you've specified the report options, click `Next` to continue to the Report destination dialog.

The Report destination dialog

You can choose the default option and select `Finish` to view the Log Analysis Expert report in the PEM client's tabbed browser, or click the radio button next to Download the report to save a copy of the report to an HTML file for later use.

If you have specified that the report should be saved to a file, the report will be downloaded.

Reviewing the Postgres Log Analysis Expert Report

If you've elected to review the report immediately, the Postgres Log Analysis Expert report will be displayed in the PEM Client window. The report header displays the date and time that the report was generated, the time period that the report spans, and the aggregation method specified when defining the report. The name of the server for which information is displayed is noted at the start of each section of the report.

The report displays the tables, graphs and charts that were selected in the Log Analysis Expert wizard. Use the `Jump To` button (located in the lower-right hand corner of the screen) to navigate to a specific graphic.

The Postgres Log Analysis Expert Report

If the report contains an analysis of more than one monitored server, charts and tables will be displayed in sets; first the graphs, tables and charts that display statistics for one server, then the graphics for the next server in the report.

7.8 SQL Profiling and Analysis

SQL profiler

Most RDBMS experts agree that inefficient SQL code is the leading cause of most database performance problems. The challenge for DBAs and developers is to locate the poorly-running SQL code in large and complex systems, and then optimize that code for better performance.

The SQL Profiler component allows a database superuser to locate and optimize poorly-running SQL code. Users of Microsoft SQL Server's Profiler will find PEM's SQL Profiler very similar in operation and capabilities. SQL Profiler is installed with each Advanced Server instance; if you are using PostgreSQL, you must download the SQL Profiler installer, and install the SQL Profiler product into each managed database instance you wish to profile.

For each database monitored by SQL Profiler, you must:

1. Edit the `postgresql.conf` file; you must include the SQL Profiler library in the shared_preload_libraries configuration parameter.

   For Linux installations, the parameter value should include:

   `$libdir/sql-profiler`

   on Windows, the parameter value should include:

   `$libdir/sql-profiler.dll`

2. Create the functions used by SQL Profiler in your database. The SQL Profiler installation program places a SQL script (named sql-profiler.sql) in the `share/postgresql/contrib` subdirectory of the main PostgreSQL installation directory on Linux systems. On Windows systems, this script is located in the `share` subdirectory. You must invoke this script on the maintenance database specified when registering the server with PEM.

3. Stop and re-start the server for the changes to take effect.

Please note: if you have connected to the PEM server with the PEM client before configuring SQL Profiler, you must disconnect and reconnect with the server to enable SQL Profiler functionality. For more detailed information about installing and configuring the SQL Profiler plugin, please refer to the PEM Installation Guide, available from the EnterpriseDB website at:

http://enterprisedb.com/products-services-training/products/documentation

Creating a New SQL Trace

SQL Profiler captures and displays a specific SQL workload for analysis in a SQL trace. You can start and review captured SQL traces immediately, or save captured traces for review at a later time. You can use SQL Profiler to create and store up to 15 named traces; use menu options to create and manage traces.

Creating a Trace

You can use the `Create trace...` dialog to define a SQL Trace for any database on which SQL Profiler has been installed and configured. installed and configured. To access the dialog, highlight the name of the database in the PEM client tree control; navigate through the Management menu to the SQL Profiler pull-aside menu, and select Create trace....

The Trace options tab

Use the fields on the `Trace options` tab to specify details about the new trace:

- Provide a name for the trace in the Name field.
- Click in the `User filter` field to specify the roles whose queries will be included the trace; optionally, check the box next to Select All to include queries from all roles.
- Click in the `Database filter` field to specify which databases to trace; optionally, check the box next to Select All to include queries against all databases.
- Specify a `trace size in the Maximum Trace File Size` field; SQL Profiler will terminate the trace when it reaches approximately the size specified.
- Specify Yes in the `Run Now` field to start the trace when you select the Create button; select No to enable fields on the Schedule tab.

The Create trace Schedule tab

Use the fields on the `Schedule` tab to specify scheduling details for the new trace:

- Use the `Start time` field to specify the starting time for the trace.
- Use the `End time` field to specify the ending time for the trace.
- Specify Yes in the `Repeat?` field to indicate that the trace should be repeated every day at the times specified; select No to enable fields on the Periodic job options tab.

The Create trace Periodic job options tab

Fields on the `Periodic job options` tab specify scheduing details about a recurring trace. Use fields in the Days section to specify the days on which the job will execute:

- Click in the `Week days` field to select the days of the week on which the trace will execute.
- Click in the `Month days` field to select the days of the month on which the trace will execute.
- Click in the `Months` field to select the months in which the trace will execute.

Use fields in the `Times` section to specify a time schedule for the trace execution:

- Click in the `Hours` field to select the hours at which the trace will execute.
- Click in the `Minutes` field to select the hours at which the trace will execute.

When you've completed the `Create trace...` dialog, click `Create` to start the newly defined trace or to schedule the trace for a later time.

The SQL Profiler tab, displaying the trace results

If you elect to execute the trace immediately, the trace results will display in the PEM client.

Opening an Existing Trace

To view a previous trace, highlight the name of the profiled database in the PEM client tree control; navigate through the `Management` menu to the SQL Profiler pull-aside menu, and select `Open trace....` You can also use the `SQL Profiler toolbar` menu to open a trace; select the `Open trace...` option. The Open trace... dialog opens.

Opening an existing trace

Highlight an entry in the trace list and click Open to open the selected trace. The selected trace opens in the SQL Profiler tab.

Filtering a Trace

A filter is a named set of (one or more) rules, each of which can hide events from the trace view. When you apply a filter to a trace, the hidden events are not removed from the trace, but are merely excluded from the display.

Click the Filter icon to open the `Trace Filter` dialog and create a rule (or set of rules) that define a filter. Each rule will screen the events within the current trace based on the identity of the role that invoked the event, or the query type invoked during the event.

To open an existing filter, select the `Open` button; to define a new filter, click the `Add (+)` icon to add a row to the table displayed on the General tab and provide rule details:

- Use the `Type` drop-down listbox to specify the trace field that the filter rule will apply to.
- Use the `Condition` drop-down listbox to specify the type of operator that SQL Profiler will apply to the Value when it filters the trace:
  - Select `Matches to` filter events that contain the specified Value.
  - Select `Does not match` to filter events that do not contain the specified Value.
  - Select `Is equal to` to filter events that contain an exact match to the string specified in the Value field.
  - Select `Is not equal` to to filter events that do not contain an exact match to the string specified in the Value field.
  - Select `Starts with` to filter events that begin with the string specified in the Value field.
  - Select `Does not start with` to filter events that do not begin with the string specified in the Value field.
  - Select `Less than` to filter events that have a numeric value less than the number specified in the Value field.
  - Select `Greater than` to filter events that have a numeric value greater than the number specified in the Value field.
  - Select `Less than or equal to` to filter events that have a numeric value less than or equal to the number specified in the Value field.
  - Select `Greater than or equal to` to filter events that have a numeric value greater than or equal to the number specified in the Value field.
- Use the `Value` field to specify the string, number or regular expression that SQL Profiler will search for.

When you've finished defining a rule, click the Add (+) icon to add another rule to the filter. To delete a rule from a filter, highlight the rule and click the Delete icon.

Click the `Save` button to save the filter definition to a file without applying the filter; to apply the filter, click `OK` . Select `Cancel` to exit the dialog and discard any changes to the filter.

Deleting a Trace

To delete a trace, highlight the name of the profiled database in the PEM client tree control; navigate through the `Management` menu to the SQL Profiler pull-aside menu, and select `Delete trace(s)....` You can also use the SQL Profiler toolbar menu to delete a trace; select the `Delete trace(s)...` option. The `Delete traces` dialog opens.

The Delete traces… dialog

Click the icon to the left of a trace name to mark one or more traces for deletion and click `Delete` . The PEM client will acknowledge that the selected traces have been deleted.

Viewing Scheduled Traces

To view a list of scheduled traces, highlight the name of the profiled database in the PEM client tree control; navigate through the `Management` menu to the SQL Profiler pull-aside menu, and select `Scheduled traces...` You can also use the SQL Profiler toolbar menu to the list; select the `Scheduled traces...` option.

Reviewing scheduled traces

The `Scheduled traces...` dialog displays a list of the traces that are awaiting execution. Click the edit button to the left of a trace name to access detailed information about the trace:

- The `Status` field lists the status of the current trace.
- The `Enabled?` switch displays Yes if the trace is enabled; No if it is disabled.
- The `Name` field displays the name of the trace.
- The `Agent` field displays the name of the agent responsible for executing the trace.
- The `Last run` field displays the date and time of the last execution of the trace.
- The `Next run` field displays the date and time of the next scheduled trace.
- The `Created` field displays the date and time that the trace was defined.

Using the Index Advisor

Index Advisor is distributed with Advanced Server 9.0 and above. Index Advisor works with SQL Profiler by examining collected SQL statements and making indexing recommendations for any underlying tables to improve SQL response time. The Index Advisor works on all DML (INSERT, UPDATE, DELETE) and SELECT statements that are invoked by a superuser.

Diagnostic output from the Index Advisor includes:

- Forecasted performance benefits from any recommended indexes
- The predicted size of any recommended indexes
- DDL statements you can use to create the recommended indexes

Before using Index Advisor, you must:

1. Modify the `postgresql.conf` file on each Advanced Server host, adding the index_advisor library to the shared_preload_libraries parameter.
2. Install the `Index Advisor contrib` module. To install the module, use the psql client or PEM Query Tool to connect to the database, and invoke the following command:

   ```
   \i <complete_path>/share/contrib/index_advisor.sql
   ```

3. Restart the server for your changes to take effect.

Index Advisor can make indexing recommendations based on trace data captured by SQL Profiler. Simply highlight one or more queries in the `SQL Profiler Trace Data` pane, and click the `Index Advisor` toolbar button (or select Index Advisor from the View menu). For detailed usage information about Index Advisor, please see the EDB Postgres Advanced Server Guide.

Please note: Index Advisor cannot analyze statements invoked by a non-superuser. If you attempt to analyze statements invoked by a non-superuser, the server log will include the following error:

```
ERROR: access to library "index_advisor" is not allowed
```

For more information about configuring and using Index Advisor, please see the EDB Postgres Advanced Server Guide, available from EnterpriseDB at:

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

7.9 Tuning Wizard

Tuning Wizard

The Tuning Wizard reviews your PostgreSQL or Advanced Server installation, and recommends a set of configuration options that will help tune the installation to best suit its anticipated workload. Please note that benchmarking systems or systems with a high work load may require additional manual tuning to reach optimum performance.

Before using the Tuning Wizard, you must specify the name of the service in the Service ID field on the Advanced tab of the server's Properties dialog. PEM will use the service name when restarting the service after tuning.

The Tuning Wizard can only make recommendations for those servers that reside on the same server as their bound PEM agent. If you have specified a value of Yes in the Remote monitoring field when defining your server, the server will not be displayed in the Tuning Wizard tree control.

To open the Tuning Wizard, select `Tuning Wizard...` from the `Management` menu of the PEM client. The Tuning Wizard opens, welcoming you.

The Tuning Wizard Welcome dialog

Click `Next` to continue to the server selection dialog.

The Select Servers dialog

Expand the `Servers` node of the tree control to view a list of the servers that are currently monitored by PEM that are available for tuning. Check a box to the left of a server name to select the server for tuning.

Note

the Tuning Wizard displays a red warning symbol to the left of a server name in the tree control if the service name for that server is not provided on the server's Properties dialog.

Click `Next` to continue to the Configuration dialog.

The Configuration dialog

Select an option in the `Machine utilization` field to specify the type of work performed by the selected servers. The type of work performed by the server determines how the tuning wizard will allocate system resources:

- Select `Dedicated` to dedicate the majority of the system resources to the database server.
- Select `Mixed use` to dedicate a moderate amount of system resources to the database server.
- Select `Developer workstation` to dedicate a relatively small amount of system resources to the database server.

Select an option in the `Workload Selection` field to specify the type of workload typically performed on the selected server:

- Select `OLTP` if the selected server is used primarily to process online transaction workloads.
- Select `Mixed` if the selected server provides a mix of transaction processing and data reporting.
- Select `Data warehouse` if the server is used for heavy data reporting.

Click `Next` to continue to the `Tuning Changes Summary` dialog.

The Tuning Changes Summary dialo

The tree control on the `Tuning Changes Summary` dialog displays the parameter setting modifications recommended for each server analyzed by the Tuning Wizard. Use the checkboxes next to a server or parameter name to select the recommendations that tuning wizard will either include in a preview report or apply:

- A checked box to the left of a parameter name specifies that the Tuning Wizard will include the parameter setting.
- A checked box to the left of a server name specifies that the Tuning Wizard will include all parameter setting recommendations for the specified server.

Specify which Tuning Wizard recommendations you wish to include in a report or apply, and click `Next` to continue.

Use the `Schedule or Run?` dialog to either specify a time that PEM will apply the changes, or generate a report that details the recommended changes.

The selected actions will apply to all of the changes noted on the Tuning Changes Summary. If you opt to generate a report, PEM will create a report that contains a list of the current values and recommended modifications to the configuration parameters selected on the Tuning Changes Summary dialog. Note that to implement changes, you will need to invoke the Tuning Wizard a second time, specifying the parameters you wish to modify on the `Tuning Changes Summary` dialog.

Select `Schedule changes` to view and specify your scheduling options.

The Schedule or Run? dialog

You can:

- Set the `Configuration now?` slider to `Yes` to apply the tuning wizard's recommendations and restart the server now.
- Set the `Configuration now?` slider to `No` to enable the Time? field and use the calendar selector to specify a time for PEM to apply the tuning wizard's recommendations and restart the server. Note that if you schedule a time for the changes to be applied, you will not be provided with a preview of the change recommendations.

Select `Generate report` to view your report options.

The Schedule or Run? dialog

You can:

- Set the `View report now?` slider to `Yes` to display the Tuning Wizard report onscreen.
- Set the `View report now?` slider to `No` to enable the `Save the report to file` field and use the calendar selector to specify a file name and location to which PEM will write the Tuning Wizard report.

Click the `Finish` button to either apply the Tuning Wizard's modifications or generate a report and exit the Tuning Wizard.

The Tuning Wizard report

You can confirm that Tuning Wizard has implemented the recommended changes by reviewing the `postgresql.conf` file for the modified server. The Tuning Wizard adds a comment above each modified parameter in the postgresql.conf file when the change is applied.

Confirming a change in the postgresql.conf file

You can also confirm a parameter value by querying the server. For example, to confirm the value of the shared_buffers parameter, open a SQL command line using either the Query Tool (accessed through the Tools menu) or the psql client, and issue the command:

```
SHOW shared_buffers;
```

The value returned by the server will confirm that the parameter has been modified.

7.10 Postgres Expert - Best Practice Enforcement

Postgres Expert

The Postgres Expert utility provides expert advice on how to best configure your Postgres servers for optimal performance, security, and more. Postgres Expert serves as a PostgreSQL 'DBA in a box' by analyzing your servers for deviations in best practices. Postgres Expert contains three specialized Experts:

- The Configuration Expert.
- The Schema Expert.
- The Security Expert.

You can select specific rules for each Expert to analyze, or accept all rules, and then review Postgres Expert reports detailing any best practice issues that require your attention.

Using the Postgres Expert Wizard

To use the Postgres Expert wizard select the `Postgres Expert` option from the `Management` menu in the PEM client. When the wizard's `Welcome` window opens, click `Next` to continue.

The Postgres Expert Welcome dialog

The wizard displays a tree control that allows you to choose the Experts and Rules with which Postgres Expert will evaluate the specified server or database.

The PEM Agent Installer's Welcome dialog

The tree control categorizes the rules under three Expert headings:

- Select from the `Configuration Expert rules` to analyze the parameter settings of the server or operating system to find any adjustments that might improve system performance.
- Select from the `Schema Expert rules` to analyze schema objects (locating missing primary keys, foreign keys without indexes, etc).
- Select from the `Security Expert rules` to review the system to find security vulnerabilities.

Use the checkmark indicator to the left of an expert or rule to indicate that the Postgres Expert should analyze the configuration of the selected servers for any best practice deviations related to the checked item.

You can:

- Check the box next to the name of an expert to select or deselect all of the configuration items listed under that node of the tree control.
- Check the box next to `Servers/Databases` to instruct Postgres Expert to review the selected server for all of the items in the tree control.
- Deselect the box next to `Servers/Databases` to to un-check all of the rules; then, navigate through the tree control, specifying only the items that you wish Postgres Expert to evaluate.

After making your selections, click `Next` to continue to the Server/Databases tree control.

The Servers/Databases dialog

Select or de-select the servers and databases that you would like Postgres Expert to analyze. If you select multiple servers or databases, the resulting report will contain a separate analysis of each target. When you've finished, click `Next` to select a report destination.

Specify a report destination

You can select the default option and click `Finish` to view an onscreen report from Postgres Expert, or check the box next to Download the report to save a copy of the report to an HTML file for later use. If you choose to save the report to a file, the download will begin immediately. The file will be saved in your default download directory.

Reviewing Postgres Expert Recommendations

Postgres Expert produces an easily navigated report that contains an analysis of the selected rules, categorized by high, medium, and low severities, for the selected servers.

The Postgres Expert report

The report header contains a summary of the report, and includes the date and time that the report was generated, the number of rules analyzed, and the number of deviations from best practices found by Postgres Expert. Use the Jump to drop-down listbox to select a server to navigate to the section of the report that targets recommendations for that server.

The body of the report contains the detailed findings for each server selected for analysis. The findings are sorted by Expert; within each Expert heading, any rule violations are ranked by Severity.

The detailed recommendation for a rule

Click on each rule in the Postgres Expert report to display details and recommendations for that rule. Within each rule, section headings display:

- The `Advisor` section lists the name of the Postgres Expert advisor that prompted the recommendation.
- The `Trigger` section displays a description of the rule that raised the alert.
- The `Recommended Value` section displays the value to which Postgres Expert recommends setting the selected parameter.
- The `Description` section displays information and advice about the parameter that caused the alert.
- The `Current Values` section displays the current value(s) of any parameter(s) that influence the Postgres Expert's evaluation.

7.11 Reports

You can generate the System Configuration report and Core Usage report for all locally and remotely managed servers. To generate this report, select *Reports* from the *Management* Menu.

Reports has following options:

- System Configuration Report (JSON)
- System Configuration Report (HTML)
- Core Usage Report (JSON)
- Core Usage Report (HTML)

Please note that only superusers or the users with the pem_admin role permission can download the System Configuration or Core Usage reports.

Also note that information in these reports will reflect the latest probe run time.

System Configuration Report

The System Configuration Report provides detailed information about the PEM Agents group, PEM Server directory group and custom groups listed under browser tree. These groups can contain Postgres Enterprise Manager, PEM Agents and Database servers. You can download this report in HTML as well as in JSON format.

The *Postgres Enterprise Manager Summary* provides details about:

- The Postgres Enterprise Manager backend database server version
- Application version
- User name accessing the application
- Python version
- Flask version
- Platform specific information

The *Summary* provides information about the number of agents and servers.

System Configuration Report - PEM Summary and Summary

The *Group: PEM Agents* panel provides details about the PEM agent, CPU cores, Disk Utilization, and Memory information.

System Configuration Report - PEM Agents

The *Group: PEM Server Directory*, provides details about:

- Database server version
- Host
- Port
- Database name
- Database size
- Tablespace size

System Configuration Report - Group Server Name

Please note that here `Group Server Name` depends on the group name to which the server is added.

Core Usage Report

The Core Usage report provides detailed information about number of cores specific to:

- The server type
- Database version
- Platform and group name

The report also gives detailed information about locally managed servers:

- Type
- Host
- Port
- Platform

- Cores
- RAM

Core Usage Report

7.12 Monitoring Failover Manager

Monitoring Failover Manager

If you are using EDB Failover Manager to monitor your replication scenario, you must manually install and configure Failover Manager. For detailed information about installing Failover Manager, visit the EnterpriseDB website at:

https://www.enterprisedb.com/products/edb-postgres-platform/edb-postgres-failover-manager

To monitor the status of a Failover Manager cluster on the Streaming Replication dashboard, you must provide the following information on the `Advanced` tab of the `server Properties` dialog for each node of the cluster:

- Use the `EFM Cluster Name` field to specify the name of the Failover Manager cluster. The cluster name is the prefix of the name of the cluster properties file. For example, if your cluster properties file is named efm.properties, your cluster name is efm.
- Use the `EFM Installation Path` field to specify the location of the Failover Manager binary file. By default, the Failover Manager binary file is installed in /usr/efm-2.1/bin.

After registering your servers, the `Streaming Replication Analysis` dashboard will display status information about your EFM cluster near the bottom of the dashboard.

The Failover Manager cluster status report

The `Failover Manager Cluster Status` section of the Streaming Replication Analysis dashboard displays information about the monitored cluster:

The `Failover Manager Cluster Information` table provides information about the Failover Manager cluster:

- The `Properties` column displays the name of the cluster property.
- The `Values` column displays the current value of the property.

The `Failover Manager Node Status` table displays information about each node of the Failover Manager cluster:

- The `Agent Type` column displays the type of agent that resides on the node; the possible values are Master, Standby, Witness, Idle, and Promoting.
- The `Address` column displays the IP address of the node.
- The `Agent` column displays the status of the agent that resides on the node.
- The `DB` column displays the status of the database that resides on the node.
- The `XLog Location` column displays the transaction log location of the database.
- The `Status Information` column displays any error-related information about the node.
- The `XLog` Information column displays any error-related information about the transaction log.
- The `VIP` column displays the VIP address that is associated with the node.
- The `VIP Status` column displays True if the VIP is active for the node, False if the VIP is not.

Replacing a Master Node

You can use the PEM client to replace the Master node of a Failover Manager cluster with a standby node. To initiate the failover process, select Replace Cluster Master from the Management menu. A dialog opens, asking you to confirm that you wish to replace the current master node.

Replacing the Master node of a cluster

Select `Yes` to remove the current master node from the Failover Manager cluster and promote a standby node to the role of read/write master node within a Failover Manager cluster. The node with the highest promotion priority (defined in Failover Manager) will become the new master node. PEM will display a dialog, reporting the job status.

Confirmation of the promotion

When the job completes and the Streaming Replication Analysis dashboard refreshes, you can review the `Failover Manager Node Status` table to confirm that a standby node has been promoted to the role of master within the Failover Manager cluster.

7.13 Monitoring an xDB Replication Cluster

Monitoring an xDB Replication Cluster

Before configuring PEM to retrieve statistics from an Advanced Server or PostgreSQL database that is part of an xDB replication scenario, you must manually install and configure xDB Replication. For more information about xDB replication solutions and documentation, please visit the EnterpriseDB website at:

http://www.enterprisedb.com/products-services-training/products-overview/xdb-replication-server-multi-master

The PEM xDB Replication probe monitors lag data for clusters that use xDB multi- master or single-master replication that have a publication database that is an EDB Postgres Advanced Server or PostgreSQL database. Please note that if you have configured replication between other proprietary database hosts (i.e. Oracle or SQL Server) and Advanced Server or PostgreSQL, the probe cannot return lag information.

The Manage Probes tab

By default, the `xDB Replication` probe is disabled. To enable the `xDB Replication` probe, right click on the name of the server, and select `Connect` from the context menu; if prompted, provide authentication information. After connecting, expand the server node of the tree control, and highlight the name of the replicated database. Then, select `Manage Probes...` from the `Management` menu.

Use fields on the `Manage Probes` tab to configure the xDB Replication probe:

- Move the `Default` slider to `No` to modify the Minutes and Seconds between probe executions.
- Use the `Enabled?` slider to instruct PEM to execute the xDB Replication probe.
- Set the `Default` slider in the `Data Retention` field to No to modify the number of days that PEM will store the information retrieved by the probe.

After enabling the probe, you can use the metrics returned to create custom charts and dashboards in the PEM client.

7.14 Performance Diagnostics

Performance Diagnostic

You can use the Performance Diagnostic dashboard to analyze the database performance for Advanced Server instances by monitoring the wait events. To display the diagnostic charts, PEM uses the data collected by Advanced Server's EDB Wait States module.

For more information about EDB Wait States, see the *EDB Postgres Advanced Server Guide*, available at:

https://www.enterprisedb.com/docs/en/11.0/EPAS_Guide_v11/EDB_Postgres_Advanced_Server_Guide.1.077.html

You can analyze the Wait States data on multiple levels by narrowing down your selection of data. Each level of the chart is populated on the basis of your selection of data at the higher level.

Prerequisites

- You must have superuser privileges to access the Performance Diagnostic dashboard.

- You must ensure that the EDB Wait States module is installed. Modify the `postgresql.conf` file, adding the `edb_wait_states` library to the list of libraries in the `shared_preload_libraries` parameter:

  `shared_preload_libraries = '$libdir/edb_wait_states'`

  Then, restart the database server, and create the extension:

  `CREATE EXTENSION edb_wait_states;`

  The console will display an error if you do not meet the prerequisites.

The prerequisites error

To open the Performance Diagnostic dashboard, select the `Performance Diagnostic` option from the `Management` menu of the PEM client.

By default, the top most Performance Diagnostic chart pulls the data for one hour, starting from current date and time. The default range selection in hours in the first chart can be customized in the `Performance Diagnostic` section of the `Preferences` dialog under the `File` menu. You can also use the `Preferences` dialog to display `Performance Diagnostic` in a new browser tab. Use `Open in New Browser Tab?` to display the `Performance Diagnostics` dashboard in a new browser tab.

The Performance Diagnostics chart

Use the `Last` drop-down list box to select the duration for which you want to see the chart. Select the 1 hour, 4 hours, 12 hours, or 24 hours. You can also select the date and time through which you want the data to be displayed.

Selecting a date

The first chart displays the number of wait events that have occurred over the period of time that is selected for the charts. You can narrow down the timeline in the first chart to analyze the data for a specific period. The second chart displays the total number of wait events of each type, for the timeline that you select in the first chart. You can select the specific wait event types for which you want to analyze the data.

To make differentiation easier, the graph for each wait event type is displayed in a different color. Click on `Legend` to identify the color in which a particular wait event type is displayed in the graph.

Selecting a timeline

Select a point in time on the second chart for which you wish to analyze the wait events; the `Wait Event Details` panel is populated on the basis of your selection in the second chart. The panel makes wait details available on three tabs:

- The `SQL tab` displays the list of SQL queries having wait events for the selected sample time.
- The `Users` tab displays the details of the wait events grouped by users for selected sample time.
- The `Waits` tab displays the number of wait events belonging to each wait event type for the selected sample time.

You can filter the data displayed on the three tabs, or sort the data alphabetically by clicking on a column header.

Reviewing query details

Click on the `Eye` icon in any row to display a new tab with details of the query for a particular row. This tab displays the `Query ID` and its corresponding sessions IDs in a dropdown list. Select the session ID for the query for which you want to analyze the data; the tab will display details corresponding to the selected session ID and query ID.

The `Waits information` table displays the waiting query. If the query is partially displayed, click the down arrow at the bottom of the section to view the complete SQL query.

The Waits information panel

The `Wait Types` panel displays the total number of wait events for the selected session ID and query ID. Select a time range in the first chart to analyze the data for a specific period.

The `Wait Types` bar graph displays the total number of wait event types for the duration that you select. To make differentiation easier, each wait event type is represented by a different color in the bar graph.

The wait types bar graph

Select the range for which you want to analyze the wait event types and their corresponding wait events; the `Wait Events` donut chart is populated on the basis of your selection. In the Wait Events donut chart, all the wait event types applicable to the selected duration are displayed in the percentage format. You can select any one wait event type to see all the wait events belonging to that particular wait event type and their count. Click Read More to read about the various wait event types and wait events.

The wait events donut chart

In the `Wait Events` donut chart, all the wait event types applicable to the selected timeline are displayed a percentage format. You can select any one wait event type to see all the wait events belonging to that particular wait event type and their count. Click the `Read More` link to read about the various wait event types and wait events.

7.15 Reference

Reference

The following sections are provided for reference; please note that the items referred to in the following tables are subject to change.

PEM Server Configuration Parameters - Reference

You can use global configuration options to modify aspects of the PEM Server's behavior. Please note that the list of configuration parameters is subject to change.

Parameter name	Value and Unit
audit_log_retention_time	30 days
auto_create_agent_alerts	true
auto_create_server_alerts	true
chart_disable_bullets	false
cm_data_points_per_report	50
cm_max_end_date_in_years	5 years
dash_alerts_timeout	60 seconds
dash_db_comrol_span	7 days
dash_db_comrol_timeout	1800 seconds
dash_db_connovervw_timeout	300 seconds
dash_db_eventlag_span	7 days
dash_db_eventlag_timeout	1800 seconds
dash_db_hottable_rows	25 rows
dash_db_hottable_timeout	300 seconds
dash_db_io_span	7 days
dash_db_io_timeout	1800 seconds
dash_db_rowact_span	7 days
dash_db_rowact_timeout	1800 seconds
dash_db_storage_timeout	300 seconds
dash_db_timelag_span	7 days
dash_db_timelag_timeout	1800 seconds
dash_db_useract_span	7 days
dash_db_useract_timeout	1800 seconds
dash_efm_timeout	300 seconds
dash_global_overview_timeout	30 seconds
dash_header_timeout	60 seconds

Parameter name	Value and Unit
dash_io_chkpt_span	7 days
dash_io_chkpt_timeout	1800 seconds
dash_io_hotindx_timeout	300 seconds
dash_io_hottbl_timeout	300 seconds
dash_io_index_objectio_rows	25 rows
dash_io_index_objectio_timeout	60 seconds
dash_io_objectio_rows	25 rows
dash_io_objectio_timeout	300 seconds
dash_memory_hostmemact_span	7 days
dash_memory_hostmemact_timeout	1800 seconds
dash_memory_hostmemconf_timeout	300 seconds
dash_memory_servmemact_span	7 days
dash_memory_servmemact_timeout	1800 seconds
dash_memory_servmemconf_timeout	300 seconds
dash_objectact_objstorage_rows	15 rows
dash_objectact_objstorage_timeout	300 seconds
dash_objectact_objtopindexes_timeout	300 seconds
dash_objectact_objtoptables_timeout	300 seconds
dash_os_cpu_span	7 days
dash_os_cpu_timeout	1800 seconds
dash_os_data_span	7 days
dash_os_disk_span	7 days
dash_os_hostfs_timeout	1800 seconds
dash_os_io_timeout	1800 seconds
dash_os_memory_span	7 days
dash_os_memory_timeout	1800 seconds
dash_os_packet_span	7 days
dash_os_packet_timeout	1800 seconds
dash_os_process_span	7 days
dash_os_process_timeout	1800 seconds
dash_os_storage_timeout	1800 seconds
dash_os_traffic_span	7 days
dash_os_traffic_timeout	1800 seconds
dash_os_util_timeout	1800 seconds
dash_probe_log_timeout	300 seconds
dash_replication_archivestat_span	7 days
dash_replication_archivestat_timeout	1800 seconds
dash_replication_pagelag_span	7 days
dash_replication_pagelag_timeout	1800 seconds
dash_replication_segmentlag_span	7 days
dash_replication_segmentlag_timeout	1800 seconds
dash_replication_timelag_span	7 days
dash_replication_timelag_timeout	1800 seconds
dash_server_buffers_written	168 hours
dash_server_buffers_written_timeout	300 seconds
dash_server_connovervw_timeout	300 seconds
dash_server_database_timeout	300 seconds
dash_server_dbsize_span	7 days
dash_server_dbsize_timeout	1800 seconds
dash_server_disk_timeout	1800 seconds
dash_server_global_span	7 days
dash_server_sharedbuff_span	7 days
dash_server_sharedbuff_timeout	1800 seconds
dash_server_tabspacesize_span	7 days
dash_server_tabspacesize_timeout	1800 seconds
dash_server_useract_span	7 days
dash_server_useract_timeout	1800 seconds
dash_sessact_lockact_timeout	300 seconds
dash_sessact_workload_timeout	300 seconds

Parameter name	Value and Unit
dash_sess_waits_nowaits_timeout	300 seconds
dash_sess_waits_timewait_timeout	300 seconds
dash_sess_waits_waitdtl_timeout	300 seconds
dash_storage_dbdtls_timeout	300 seconds
dash_storage_dbovervw_timeout	300 seconds
dash_storage_hostdtls_timeout	300 seconds
dash_storage_hostovervw_timeout	300 seconds
dash_storage_tblspcdtls_timeout	300 seconds
dash_storage_tblspcovervw_timeout	300 seconds
dash_sys_waits_nowaits_timeout	300 seconds
dash_sys_waits_timewait_timeout	300 seconds
dash_sys_waits_waitdtl_timeout	300 seconds
deleted_charts_retention_time	7 days
deleted_probes_retention_time	7 days
download_chart_format	jpeg
flapping_detection_state_change	3
job_retention_time	30 days
long_running_transaction_minutes	5 minutes
nagios_cmd_file_name	<file_name>
nagios_enabled	t
nagios_medium_alert_as_critical	f
nagios_spool_retention_time	7 days
probe_log_retention_time	30 days
reminder_notification_interval	24 hours
server_log_retention_time	30 days
show_data_tab_on_graph	false
smtp_authentication	false
smtp_enabled	true
smtp_encryption smtp_password	false
smtp_port	25
smtp_server	127.0.0.1
smtp_spool_retention_time smtp_username	7 days
snmp_community	public
snmp_enabled	true
snmp_port	162
snmp_server	127.0.0.1
snmp_spool_retention_time snmp_security_name snmp_security_engine_id	7 days
snmp_security_level snmp_context_name snmp_context_engine_id	NOAUTH_NOPRIV
snmp_authentication_protocol	NONE
snmp_privacy_protocol snmp_authentication_password snmp_privacy_password	NONE
webclient_help_pg	EnterpriseDB hosted documentation

Capacity Manager Metrics - Reference

Please Note that the Capacity Manager metrics available will vary by platform, and are subject to change. The available metrics may include the metrics described in the table below.

Metric Name	Description
# Dead Tuples	The number of dead tuples in the selected table.
# Dead Tuples+	The cumulative number of dead tuples in the selected table.
# Heap Tuples Fetched by Index Scans	The number of heap tuples fetched by index scans.
# Heap Tuples Fetched by Index Scans	The cumulative number of heap tuples fetched by index scans.
# Idle Backends+	The cumulative number of currently idle backend clients.
# Index Scans	The number of index scans performed on the specified object.
# Index Scans+	The cumulative number of index scans performed on the specified object.
# Index Tuples Read	The number of index tuples read.
# Index Tuples Read+	The cumulative number of index tuples read.
# Live Tuples	The number of tuples visible to transactions.

Metric Name	Description
# Live Tuples+	The cumulative number of tuples visible to transactions.
# Pages Estimated by ANALYZE	The number of pages estimated by ANALYZE.
# Pages Estimated by ANALYZE+	The cumulative number of pages estimated by ANALYZE.
# Sequential Scans	The number of sequential scans performed on the specific table.
# Sequential Scans+	The cumulative number of sequential scans performed on the specific table.
# Sequential Scan Tuples	The number of tuples sequentially scanned in the specific table.
# Sequential Scan Tuples+	The cumulative number of tuples sequentially scanned in the specific table.
# Tuples Deleted	The number of tuples deleted.
# Tuples Deleted+	The cumulative number of tuples deleted.
# Tuples Estimated by ANALYZE	The number of live (visible) tuples estimated by ANALYZE.
# Tuples Estimated by ANALYZE+	The cumulative number of live tuples estimated by ANALYZE.
# Tuples HOT Updated	The number of tuples HOT updated. In a HOT update, the new tuple resides in t
# Tuples HOT Updated+	The cumulative number of tuples HOT updated.
# Tuples Inserted	The number of tuples inserted into the specified table.
# Tuples Inserted+	The cumulative number of tuples inserted into the specified table.
# Tuples Updated	The number of tuples updated in the selected table.
# Tuples Updated+	The cumulative number of tuples updated in the selected table.
Blocks Hit	The number of blocks found in the cache.
Blocks Hit+	The cumulative number of blocks found in the cache.
Blocks Read	The number of blocks read.
Blocks Read+	The cumulative number of blocks read.
Blocks Read from InfiniteCache	The number of blocks read from InfiniteCache.
Blocks Read from InfiniteCache+	The cumulative number of blocks read from InfiniteCache.
Blocks Written	The number of blocks written.
Blocks Written+	The cumulative number of blocks written.
Buffers Allocated	The number of buffers allocated.
Buffers Allocated+	The cumulative number of buffers allocated.
Buffers Written - Backends	The number of buffer blocks written to disk by server processe (processes conne
Buffers Written - Backends+	The cumulative number of buffer blocks written to disk by server processes.
Buffers Written - Checkpoint	The number of blocks written to disk by the checkpoint process.
Buffers Written - Checkpoint+	The cumulative number of blocks written to disk by the checkpoint process.
Buffers Written - Cleaning Scan	The number of blocks written to disk by the autovacuum process.
Buffers Written - Cleaning Scan+	The cumulative number of blocks written to disk by the autovacuum process.
Bytes Received (KB)	The number of bytes received from the client (in kilobytes).
Bytes Received (KB)+	The cumulative number of bytes received (in kilobytes).
Bytes Sent (KB)	The number of bytes sent to the client (in kilobytes).
Bytes Sent (KB)+	The cumulative number of bytes sent (in kilobytes).
Checkpoints - Timed	The number of checkpoint operations triggered by the checkpoint interval.
Checkpoints - Timed+	The cumulative number of checkpoint operations triggered by the checkpoint inte
Checkpoints - Untimed	The number of checkpoint operations triggered by checkpoint size.
Checkpoints - Untimed+	The cumulative number of checkpoint operations triggered by checkpoint size.
Database Size (MB)	The size of the specified database (in megabytes).
Free RAM Memory	The amount of free RAM memory (in megabytes).
Free Swap Memory	The amount of free swap space on disk (in megabytes).
Heap Blocks Hit	The number of heap blocks found in the cache.
Heap Blocks Hit+	The cumulative number of heap blocks found in the cache.
Heap Blocks Read	The number of heap blocks read.
Heap Blocks Read+	The cumulative number of heap blocks read.
Index Blocks Hit	The number of index blocks found in the cache.
Index Blocks Hit+	The cumulative number of index blocks found in the cache.
Index Blocks Read	The number of index blocks read.
Index Blocks Read+	The cumulative number of index blocks read.
Index Size (MB)	The size of the specified index (in megabytes).
In Packets Discards	The number of inbound packets discarded.
In Packets Discards+	The cumulative number of inbound packets discarded.
In Packets Errors	The number of inbound packets that contain errors.
In Packets Errors+	The cumulative number of inbound packets that contain errors.
Link Bandwidth (Mbit/s)	The speed of the network adapter (in megabits per second).
Load Average - 15 Minute	CPU saturation (in percent) - 15 minute sampling average.

Metric Name	Description
Load Average - 1 Minute	CPU saturation (in percent) - 1 minute sampling average.
Load Average - 5 Minute	CPU saturation (in percent) - 5 minute sampling average.
Load Percentage	CPU saturation in percent.
Number of Prepared Transactions+	The cumulative number of prepared transactions.
Number of WAL Files+	The cumulative number of write-ahead log files.
Out Packets Discards	The number of outbound packets discarded.
Out Packets Discards+	The cumulative number of outbound packets discarded.
Out Packets Errors	The number of outbound packets that contain errors.
Out Packets Errors+	The cumulative number of outbound packets that contain errors.
Packets Received	The number of packets received.
Packets Received+	The cumulative number of packets received.
Packets Sent	The number of packets sent.
Packets Sent+	The cumulative number of packets sent.
Size (MB)	The total size of the disk (in megabytes).
Size of Indexes (MB)	The size of indexes on the specified table (in megabytes).
Space Available (MB)	The current disk space available (in megabytes).
Space Used (MB)	The current disk space used (in megabytes).
Table Size (MB)	The size of the specified table (in megabytes).
Tablespace Size (MB)	The size of the specified tablespace (in megabytes).
Temp Buffers (MB)	The size of temporary buffers (in megabytes).
Toast Blocks Hit	The number of TOAST blocks found in the cache.
Toast Blocks Hit+	The cumulative number of TOAST blocks found in the cache.
Toast Blocks Read	The number of TOAST blocks read.
Toast Blocks Read+	The cumulative number of TOAST blocks read.
Total RAM Memory	The total amount of RAM memory on the system (in megabytes).
Total Swap Memory	The total amount of swap space on the system (in megabytes).
Total Table Size w/Indexes and Toast	The total size of the specified table (including indexes and associated oversized
Transactions Aborted	The number of aborted transactions.
Transactions Aborted+	The cumulative number of aborted transactions.
Transactions Committed	The number of committed transactions.
Transactions Committed+	The cumulative number of committed transactions.
Tuples Deleted	The number of tuples deleted from the specified table.
Tuples Deleted+	The cumulative number of tuples deleted from the specified table.
Tuples Estimated by ANALYZE	The number of visible tuples in the specified table.
Tuples Estimated by ANALYZE+	The cumulative number of visible tuples in the specified table.
Tuples Fetched	The number of tuples fetched from the specified table.
Tuples Fetched+	The cumulative number of tuples fetched from the specified table.
Tuples HOT Updated	The number of tuples HOT updated. In a HOT update, the new tuple resides in t
Tuples HOT Updated+	The cumulative number of tuples HOT updated. In a HOT update, the new tuple
Tuples Inserted	The number of tuples inserted into the specified table.
Tuples Inserted+	The cumulative number of tuples inserted into the specified table.
Tuples Returned	The number of tuples returned in result sets.
Tuples Returned+	The cumulative number of tuples returned in result sets.
Tuples Updated	The number of tuples updated in the specified table.
Tuples Updated+	The cumulative number of tuples updated in the specified table.
WAL Segment Size (MB)	The segment size of the write-ahead log (in megabytes).

Note

The ’+’ following the name of a metric signifies that the data for the metric is gathered cumulatively; those metrics that are not followed by the ’+’ sign are collected as a ’point-in-time’ value.

PEM Probes – Reference

A probe is a scheduled task that retrieves information about the database objects that are being monitored by the PEM agent. PEM uses the collected information to build the graphs displayed on each dashboard. The Manage Probes tab (accessed via the Management menu) allows you to modify the data collection schedule and the length of time that PEM will retain information returned by a specific probe.

PEM Pre-defined Alert Templates – Reference

An alert definition contains a system-defined or user-defined set of conditions that PEM compares to the system statistics; if the statistics deviate from the boundaries specified for that statistic, the alert triggers, and the PEM client displays a warning on the *Alerts Overview* page, and optionally sends a notification to a monitoring user.

The tables that follow list the system-defined alert templates that you can use to create an alert; please note that this list is subject to change, and may vary by system:

Templates applicable on Agent

Template Name	Description
Load Average (1 minute)	1-minute system load average.
Load Average (5 minutes)	5-minute system load average.
Load Average (15 minutes)	15-minute system load average.
Load Average per CPU Core (1 minutes)	1-minute system load average per CPU cor
Load Average per CPU Core (5 minutes)	5-minute system load average per CPU cor
Load Average per CPU Core (15 minutes)	15-minute system load average per CPU co
CPU utilization	Average CPU consumption.
Number of CPUs running higher than a	Number of CPUs running at greater than K%
Free memory percentage	Free memory as a percent of total system m
Memory used percentage	Percentage of memory used.
Swap consumption	Swap space consumed (in megabytes).
Swap consumption percentage	Percentage of swap area consumed.
Disk Consumption	Disk space consumed (in megabytes).
Disk consumption percentage	Percentage of disk consumed.
Disk Available	Disk space available (in megabytes).
Disk busy percentage	Percentage of disk busy.
Most used disk percentage	Percentage used of the most utilized disk or
Total table bloat on host	The total space wasted by tables on a host,
Highest table bloat on host	The most space wasted by a table on a host
Average table bloat on host	The average space wasted by tables on hos
Table size on host	The size of tables on host, in MB.
Database size on host	The size of databases on host, in MB.
Number of ERRORS in the logfile on agent N in last X hours.	The number of ERRORS in the logfile on ag
Number of WARNINGS in the logfile on agent N in last X hours	The number of WARNINGS in the logfile on
Number of WARNINGS or ERRORS in the logfile on agent N in last X hours	The number of WARNINGS or ERRORS in
Package version mismatch	Check for package version mismatch as per
Total materialized view bloat on host	The total space wasted by materialized view
Highest materialized view bloat on host	The most space wasted by a materialized vi
Average materialized view bloat on host	The average space wasted by materialized
Materialized view size on host	The size of materialized views on host, in M
Agent Down	Specified agent is currently down.

Templates applicable on Server

Template Name	Description
Total table bloat in server	The total space wasted by tables in ser
Largest table (by multiple of unbloated size)	Largest table in server, calculated as a
Highest table bloat in server	The most space wasted by a table in se
Average table bloat in server	The average space wasted by tables in
Table size in server	The size of tables in server, in MB.
Database size in server	The size of databases in server, in MB.
Number of WAL files	Total number of Write Ahead Log files.
Number of prepared transactions	Number of transactions in prepared sta
Total connections	Total number of connections in the serv
Total connections as percentage of max_connections	Total number of connections in the serv connections allowed on server, settings
Unused, non-superuser connections	Number of unused, non-superuser conn

Template Name	Description
Unused, non-superuser connections as percentage of max_connections	Number of unused, non-superuser conn
Ungranted locks	Number of ungranted locks in server.
Percentage of buffers written by backends	The percentage of buffers written by ba
Percentage of buffers written by checkpoint	The percentage of buffers written by the
Buffers written per second	Number of buffers written per second, o
Buffers allocated per second	Number of buffers allocated per second
Connections in idle state	Number of connections in server that ar
Connections in idle-in-transaction state	Number of connections in server that ar
Connections in idle-in-transaction state, as percentage of max_connections	Number of connections in server that ar
Long-running idle connections	Number of connections in the server tha
Long-running idle connections and idle transactions	Number of connections in the server tha
Long-running idle transactions	Number of connections in the server tha
Long-running transactions	Number of transactions in server that h
Long-running queries	Number of queries in server that have b
Long-running vacuums	Number of vacuum operations in server
Long-running autovacuums	Number of autovacuum operations in se
Committed transactions percentage	Percentage of transactions in the serve
Shared buffers hit percentage	Percentage of block read requests in th
Tuples inserted	Tuples inserted into server over last N r
InfiniteCache buffers hit percentage	Percentage of block read requests in th
Tuples fetched	Tuples fetched from server over last N r
Tuples returned	Tuples returned from server over last N
Dead Tuples	Number of estimated dead tuples in ser
Tuples updated	Tuples updated in server over last N mi
Tuples deleted	Tuples deleted from server over last N r
Tuples hot updated	Tuples hot updated in server, over last
Sequential Scans	Number of full table scans in server, ov
Index Scans	Number of index scans in server, over l
Hot update percentage	Percentage of hot updates in the server
Live Tuples	Number of estimated live tuples in serv
Dead tuples percentage	Percentage of estimated dead tuples in
Last Vacuum	Hours since last vacuum on the server.
Last AutoVacuum	Hours since last autovacuum on the se
Last Analyze	Hours since last analyze on the server.
Last AutoAnalyze	Hours since last autoanalyze on the ser
Percentage of buffers written by backends over the last N minutes	The percentage of buffers written by ba
Table Count	Total number of tables in server.
Function Count	Total number of functions in server.
Sequence Count	Total number of sequences in server.
A user expires in N days	Number of days before a user's validity
Index size as a percentage of table size	Size of the indexes in server, as a perce
Largest index by table-size percentage oc_index, table_size.	Largest index in server, calculated as p
Number of ERRORS in the logfile on server M in the last X hours	The number of ERRORS in the logfile o
Number of WARNINGS in the logfile on server M in the last X hours	The number of WARNINGS in logfile on
Number of WARNINGS or ERRORS in the logfile on server M in the last X hours	The number of WARNINGS or ERROR
Number of attacks detected in the last N minutes	The number of SQL injection attacks oc
Number of attacks detected in the last N minutes by username	The number of SQL injection attacks oc
Number of standby servers lag behind the master by write location	Streaming Replication: number of stand
Number of standby servers lag behind the master by flush location	Streaming Replication: number of stand
Number of standby servers lag behind the master by replay location	Streaming Replication: number of stand
Standby server lag behind the master by write location	Streaming Replication: standby server
Standby server lag behind the master by flush location	Streaming Replication: standby server
Standby server lag behind the master by replay location	Streaming Replication: standby server
Standby server lag behind the master by size (MB)	Streaming Replication: standby server
Standby server lag behind the master by WAL segments	Streaming Replication: standby server
Standby server lag behind the master by WAL pages	Streaming Replication: standby server
Total materialized view bloat in server	The total space wasted by materialized
Largest materialized view (by multiple of unbloated size)	Largest materialized view in server, calc
Highest materialized view bloat in server	The most space wasted by a materializ

Template Name	Description
Average materialized view bloat in server	The average space wasted by materiali
Materialized view size in server	The size of materialized view in server,
View Count	Total number of views in server.
Materialized View Count	Total number of materialized views in se
Audit config mismatch	Check for audit config parameter misma
Server Down	Specified server is currently inaccessib
Number of WAL archives pending	Streaming Replication: number of WAL
Number of minutes lag of standby server from master server	Streaming Replication: number of minu
Log config mismatch	Check for log config parameter mismat

Templates applicable on Database

Template Name	Description
Total table bloat in database	The total space wasted by tables in database
Largest table (by multiple of unbloated size)	Largest table in database, calculated as a mu
Highest table bloat in database	The most space wasted by a table in databas
Average table bloat in database	The average space wasted by tables in datab
Table size in database	The size of tables in database, in MB.
Database size	The size of the database, in MB.
Total connections	Total number of connections in the database.
Total connections as percentage of max_connections	Total number of connections in the database
Ungranted locks	Number of ungranted locks in database.
Connections in idle state	Number of connections in database that are i
Connections in idle-in-transaction state	Number of connections in database that are i
Connections in idle-in-transaction state,as percentage of max_connections	Number of connections in database that are i
Long-running idle connections	Number of connections in the database that h
Long-running idle connections and idle transactions	Number of connections in the database that h
Long-running idle transactions	Number of connections in the database that h
Long-running transactions	Number of transactions in database that have
Long-running queries	Number of queries in database that have bee
Long-running vacuums	Number of vacuum operations in database th
Long-running autovacuums	Number of autovacuum operations in databas
Committed transactions percentage	Percentage of transactions in the database th
Shared buffers hit percentage	Percentage of block read requests in the data
InfiniteCache buffers hit percentage	Percentage of block read requests in the data
Tuples fetched	Tuples fetched from database over last N mir
Tuples returned	Tuples returned from database over last N m
Tuples inserted	Tuples inserted into database over last N mir
Tuples updated	Tuples updated in database over last N minu
Tuples deleted	Tuples deleted from database over last N mir
Tuples hot updated	Tuples hot updated in database, over last N r
Sequential Scans	Number of full table scans in database, over
Index Scans	Number of index scans in database, over last
Hot update percentage	Percentage of hot updates in the database ov
Live Tuples	Number of estimated live tuples in database.
Dead Tuples	Number of estimated dead tuples in database
Dead tuples percentage	Percentage of estimated dead tuples in datab
Last Vacuum	Hours since last vacuum on the database.
Last AutoVacuum	Hours since last autovacuum on the databas
Last Analyze	Hours since last analyze on the database.
Last AutoAnalyze	Hours since last autoanalyze on the databas
Table Count	Total number of tables in database.
Function Count	Total number of functions in database.
Sequence Count	Total number of sequences in database.
Index size as a percentage of table size	Size of the indexes in database, as a percent
Largest index by table-size percentage	Largest index in database, calculated as perc
Database Frozen XID	The age (in transactions before the current tr

Template Name	Description
Number of attacks detected in the	The number of SQL injection attacks occurre
Number of attacks detected in the	The number of SQL injection attacks occurre
Queries that have been cancelled due to dropped tablespaces	Streaming Replication: number of queries tha
Queries that have been cancelled due to lock timeouts	Streaming Replication: number of queries tha
Queries that have been cancelled due to old snapshots	Streaming Replication: number of queries tha
Queries that have been cancelled due to pinned buffers	Streaming Replication: number of queries tha
Queries that have been cancelled due to deadlocks	Streaming Replication: number of queries tha
Total events lagging in all slony clusters	Slony Replication: total events lagging in all s
Events lagging in one slony cluster	Slony Replication: events lagging in one slon
Lag time (minutes) in one slony cluster	Slony Replication: lag time (minutes) in one s
Total rows lagging in xdb single master replication	xDB Replication: Total rows lagging in xdb si
Total rows lagging in xdb multi master replication	xDB Replication: Total rows lagging in xdb m
Total materialized view bloat in database	The total space wasted by materialized views
Largest materialized view (by multiple of unbloated size)	Largest materialized view in database, calcul
Highest materialized view bloat in database	The most space wasted by a materialized vie
Average materialized view bloat in database	The average space wasted by materialized v
Materialized view size in database	The size of materialized view in database, in
View Count	Total number of views in database.
Materialized View Count	Total number of materialized views in databa

Templates applicable on Schema

Template Name	Description
Total table bloat in schema	The total space wasted by tables in schema, in MB.
Largest table (by multiple of unbloated size)	Largest table in schema, calculated as a multiple of its own esti
Highest table bloat in schema	The most space wasted by a table in schema, in MB.
Average table bloat in schema	The average space wasted by tables in schema, in MB.
Table size in schema	The size of tables in schema, in MB.
Tuples inserted	Tuples inserted in schema over last N minutes.
Tuples updated	Tuples updated in schema over last N minutes.
Tuples deleted	Tuples deleted from schema over last N minutes.
Tuples hot updated	Tuples hot updated in schema, over last N minutes.
Sequential Scans	Number of full table scans in schema, over last N minutes.
Index Scans	Number of index scans in schema, over last N minutes.
Hot update percentage	Percentage of hot updates in the schema over last N minutes.
Live Tuples	Number of estimated live tuples in schema.
Dead Tuples	Number of estimated dead tuples in schema.
Dead tuples percentage	Percentage of estimated dead tuples in schema.
Last Vacuum	Hours since last vacuum on the schema.
Last AutoVacuum	Hours since last autovacuum on the schema.
Last Analyze	Hours since last analyze on the schema.
Last AutoAnalyze	Hours since last autoanalyze on the schema.
Table Count	Total number of tables in schema.
Function Count	Total number of functions in schema.
Sequence Count	Total number of sequences in schema.
Index size as a percentage of table size	Size of the indexes in schema, as a percentage of their table's s
Largest index by table-size percentage	Largest index in schema, calculated as percentage of its table's
Materialized View bloat	Space wasted by the materialized view, in MB.
Total materialized view bloat in schema	The total space wasted by materialized views in schema, in MB
Materialized view size as a multiple of unbloated size	Size of the materialized view as a multiple of estimated unbloat
Largest materialized view (by multiple of unbloated size)	Largest materialized view in schema, calculated as a multiple o
Highest materialized view bloat in schema	The most space wasted by a materialized view in schema, in M
Average materialized view bloat in schema	The average space wasted by materialized views in schema, in
Materialized view size	The size of materialized view, in MB.
Materialized view size in schema	The size of materialized views in schema, in MB.
View Count	Total number of views in schema.
Materialized View Count	Total number of materialized views in schema.

Template Name	Description
Materialized View Frozen XID	The age (in transactions before the current transaction) of the m

Templates applicable on Table

Template Name	Description
Table bloat	Space wasted by the table, in MB.
Table size	The size of table, in MB.
Table size as a multiple of ubloated size	Size of the table as a multiple of estimated unbloated size.
Tuples inserted	Tuples inserted in table over last N minutes.
Tuples updated	Tuples updated in table over last N minutes.
Tuples deleted	Tuples deleted from table over last N minutes.
Tuples hot updated	Tuples hot updated in table, over last N minutes.
Sequential Scans	Number of full table scans on table, over last N minutes.
Index Scans	Number of index scans on table, over last N minutes.
Hot update percentage	Percentage of hot updates in the table over last N minutes.
Live Tuples	Number of estimated live tuples in table.
Dead Tuples	Number of estimated dead tuples in table.
Dead tuples percentage	Percentage of estimated dead tuples in table.
Last Vacuum	Hours since last vacuum on the table.
Last AutoVacuum	Hours since last autovacuum on the table.
Last Analyze	Hours since last analyze on the table.
Last AutoAnalyze	Hours since last autoanalyze on the table.
Row Count	Estimated number of rows in a table.
Index size as a percentage of table size	Size of the indexes on table, as a percentage of table's size.
Table Frozen XID	The age (in transactions before the current transaction) of the table's frozen tran

Global Templates

Template Name	Description
Agents Down	Number of agents that haven't reported in recently.
Servers Down	Number of servers that are currently inaccessible.
Alert Errors	Number of alerts in an error state.

7.16 Conclusion

Conclusion

The goal of Postgres Enterprise Manager is provide you with a solution that allows you to intelligently manage all your database servers across your enterprise with a single console. To meet this objective, PEM supplies you with all the core features and functionality needed for visual database administration, as well as a number of advanced components that assist you in managing the performance and design of your database servers.

For more information about Postgres Enterprise Manager, please visit the EnterpriseDB Web site (http://www.enterprisedb.com) where you will find PEM's online documentation, as well as other tutorials and educational aids.

EnterpriseDB is the enterprise PostgreSQL company, providing products and services worldwide that are based on and support PostgreSQL, the world's most advanced open source database. EDB's products are ideally suited for transaction-intensive applications requiring superior performance, massive scalability, and compatibility with proprietary database products. EDB's products provide an economical open source alternative or complement to proprietary databases without sacrificing features or quality.

If you would like to discuss training, consulting, or enterprise support options, please contact EnterpriseDB. EnterpriseDB has offices in North America, Europe, and Asia. EnterpriseDB was founded in 2004 and is headquartered in Bedford, MA. For more information, please visit http://www.enterprisedb.com.

Sales Inquiries:

sales-us@enterprisedb.com (US)
sales-intl@enterprisedb.com (Intl)
+1-781-357-3390 or 1-877-377-4352 (US Only)

General Inquiries:

info@enterprisedb.com
info.asiapacific@enterprisedb.com (APAC)
info.emea@enterprisedb.com (EMEA)

EDB Postgres Enterprise Manager Enterprise Features Guide

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

- EDB designs, establishes coding best practices, reviews, and verifies input validation for the logon UI for EDB Postgres Enterprise Manager 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.