Postgres Enterprise Manager™

Getting Started Guide

April 10, 2012

Postgres Enterprise Manager™ Getting Started Guide, Version 2.1.0
by EnterpriseDB Corporation

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

Table of Contents

1 Introduction

This document provides an introduction to Postgres Enterprise Manager™ (PEM) and is written to acquaint you with the basics of the toolset and help you be successful in your database management activities. This guide is broken up into the following core sections and categories:

- **Postgres Enterprise Manager™ Overview** - Chapter 2 provides information about PEM functionality, components, architecture, and supported platforms.
- **Installation, configuration and general setup** – Chapter 3 provides an overview of PEM's installation and configuration steps.
- **General database administration** – Chapter 4 will assist you in performing general database administration tasks, and SQL query and stored procedure development.
- **Performance monitoring** – Chapter 5 provides an introduction to monitoring and troubleshooting performance on your database servers.
- **Capacity planning** – Chapter 6 discusses how to perform trend analysis and forecasting from data collected in Postgres Enterprise Manager™.
- **Audit logging** – Chapter 7 discusses how to perform auditing of Advanced Server databases for selected types of SQL statements.
- **SQL capture and profiling** – Chapter 8 describes how to capture a SQL workload and analyze the output to tune and fix poorly running SQL.
- **Postgres Expert - Best Practices Enforcement** – Chapter 9 discusses how to use Postgres Expert to uncover configuration or design issues that need correcting on your database servers.

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 offline documentation, made available by the PEM client.

1.1 Typographical Conventions Used in this Guide

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

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

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

2 Postgres Enterprise Manager™ - Overview

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 Postgres Plus 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.

2.1 Why Postgres Enterprise Manager™?

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

- **Management En-Mass 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 Logging.** PEM includes the Audit Manager, which provides the capability to set up audit logging on Postgres Plus 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.

- **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 Postgres Plus 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.

- **Wide Platform Support**. PEM supports most major Linux, Windows, and Solaris platforms.

2.2 General Architecture

PEM is composed of three primary components (see Figure 2.1):

The PEM Server

The PEM server provides the functionality at the core of Postgres Enterprise Manager™. The server is responsible for:

- Performing administration functions
- Processing information received from agents
- Maintaining information in its repository.

The PEM Agent

The PEM agent is responsible for performing tasks on each managed machine and collecting statistics for the database server and operating system.

The PEM Client

The PEM client is the user console from which all operations are carried out (e.g. database administration, viewing performance information, etc.).

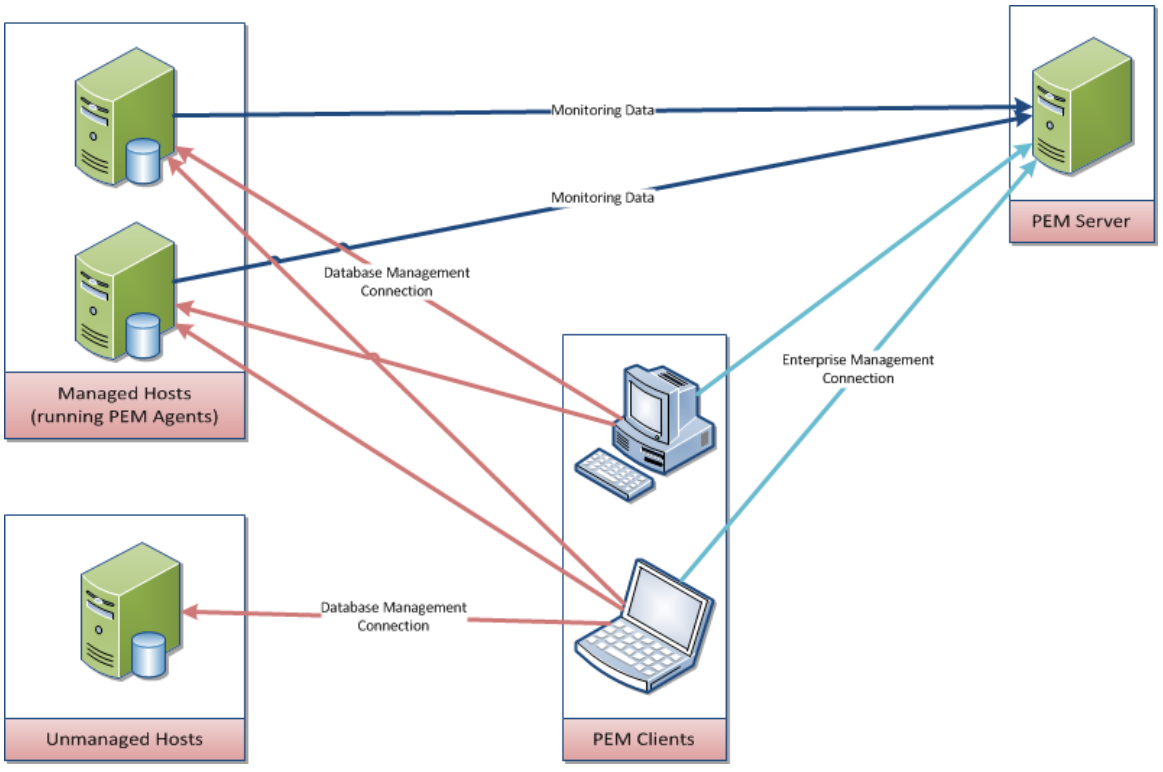

Figure 2.1 - The Postgres Enterprise Manager™ general architecture.

2.3 Supported Platforms

PEM currently provides support for the following platforms:

PEM Agent:

- Windows (32 and 64bit)
- Linux (32 and 64bit)
- Solaris
- HP-UX

PEM Server:

- Windows (32 and 64bit)
- Linux (32 and 64bit)

PEM Client:

- Windows (32 and 64bit)
- Linux (32 and 64bit)

PostgreSQL Version Support

PEM can manage and monitor:

- PostgreSQL versions 8.2 and higher
- Postgres Plus Advanced Server 8.3r2 and higher

SQL Profiler

SQL Profiler for Postgres Enterprise Manager™ is supported on EnterpriseDB distributions of PostgreSQL version 8.4 and above, and Postgres Plus Advanced Server version 9.0 and above.

3 Installing Postgres Enterprise Manager™

This section provides an overview of Postgres Enterprise Manager™ (PEM) installation and configuration; for more detailed instructions, please consult the Postgres Enterprise Manager™ Installation Guide, available at:

http://enterprisedb.com/products-services-training/products/documentation

The basic steps involved in the installation of PEM are:

1. Install the PEM server on the host server from which you will monitor the system. Installation of the server will also prompt the installation of PostgreSQL 9.0 and Apache/PHP (server prerequisites).

2. Install the PEM client on the machine from which you will manage all of your Postgres servers.

3. Install a PEM agent on each physical or virtual machine that you would like to manage with PEM.

4. Install the SQL Profiler component into each Postgres instance on which you want to perform SQL capture and analysis.

Graphical installers will lead you through installing and configuring each component of PEM.

When you invoke the installer for the PEM server, the installer performs a system check; if you already have an installation of PostgreSQL 9.0, the installer will upgrade the installation for use as the PostgreSQL Enterprise Management Server. If you do not currently have an installation, the server installer will install PostgreSQL 9.0 and Apache/PHP.

The PEM server installer will also install an agent on the PEM server machine.

You should install the PEM client on the machine from which you intend to manage your Postgres servers. The only information required by the client installer is an installation directory; the installer has no other prerequisites.

After the server is installed and running, install PEM agents on any other machine(s) that you wish to manage with PEM. The graphical installers will lead you through the very quick installation and automatically start the agents so they are up and running when the installation concludes.

The final step is to install and configure the SQL Profiler component into each Postgres instance on which you want to capture and analyze SQL workloads. The installer simply asks for the location of your PostgreSQL installation and then it places the needed software into that directory. If you are running Postgres Plus Advanced Server, the SQL Profiler plugin is already installed, and requires only configuration to enable profiling.

3.1 Starting and Stopping the PEM Server and Agents

Starting and stopping the Enterprise Manager Server is identical to controlling a Postgres server instance. The same commands that stop or start PostgreSQL on Linux and Windows control the Enterprise Manager Server.

To control the agents on a Windows platform, you can use the `Services` interface to start, stop, and set the properties (e.g. whether they auto-start or not) of your agents.

On Linux, to start an agent, use the `/etc/init.d/pemagent start` command. To stop the agent, just substitute 'stop' in place of 'start'.

On Solaris, start the agent with the `svcadm enable pemagent:default` command. To stop a Solaris agent, you use the `svcadm disable pemagent:default` command.

For more information about starting and stopping the Postgres service, please consult the Postgres Plus Advanced Server Installation Guide, available at:

http://enterprisedb.com/products-services-training/products/documentation

3.2 User Interface Basics

The main elements of the PEM client interface are the Dashboard Browser, the Main Toolbar and the Tree Control, as shown in Figure 3.1.

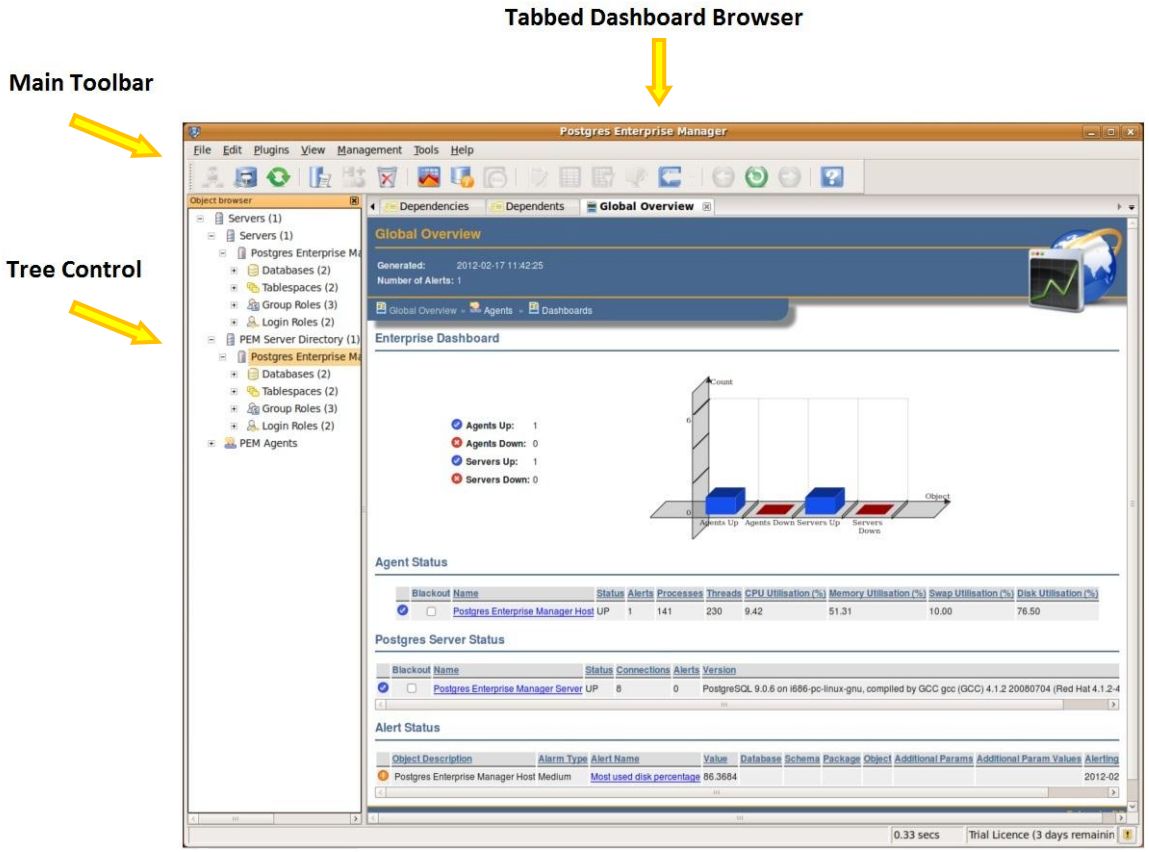

Figure 3.1 - The PEM client interface.

PEM uses a standard tree control to connect to and navigate through the contents of all managed servers. The menu bar provides easy access to various tasks, and is context sensitive so only possible tasks for selected objects in the tree control will be active. The graphical toolbar provides quick access to the most commonly used tasks and utilities.

The right pane of the client interface allows you to use tabbed browsing to review details about selected objects in the tree control. Like most web browsers, you can open multiple tabs for different views, close selected tabs when you're finished reviewing the contents, and navigate through multiple reports on the same tab using back and forward toolbar buttons.

Note that some utilities and interfaces in PEM do not run within the client interface, but instead open in a separate interactive dialog. These include interfaces such as the SQL

IDE and the SQL Profiler. This allows for greater flexibility when using these tools on systems with multiple monitors, or virtual desktops.

3.3 Online Help and Documentation

PEM contains built-in help that you can reference for assistance in using the tool (see Figure 3.2). To access the full online help for PEM, select `Help contents` from the `Help` option on the main menu.

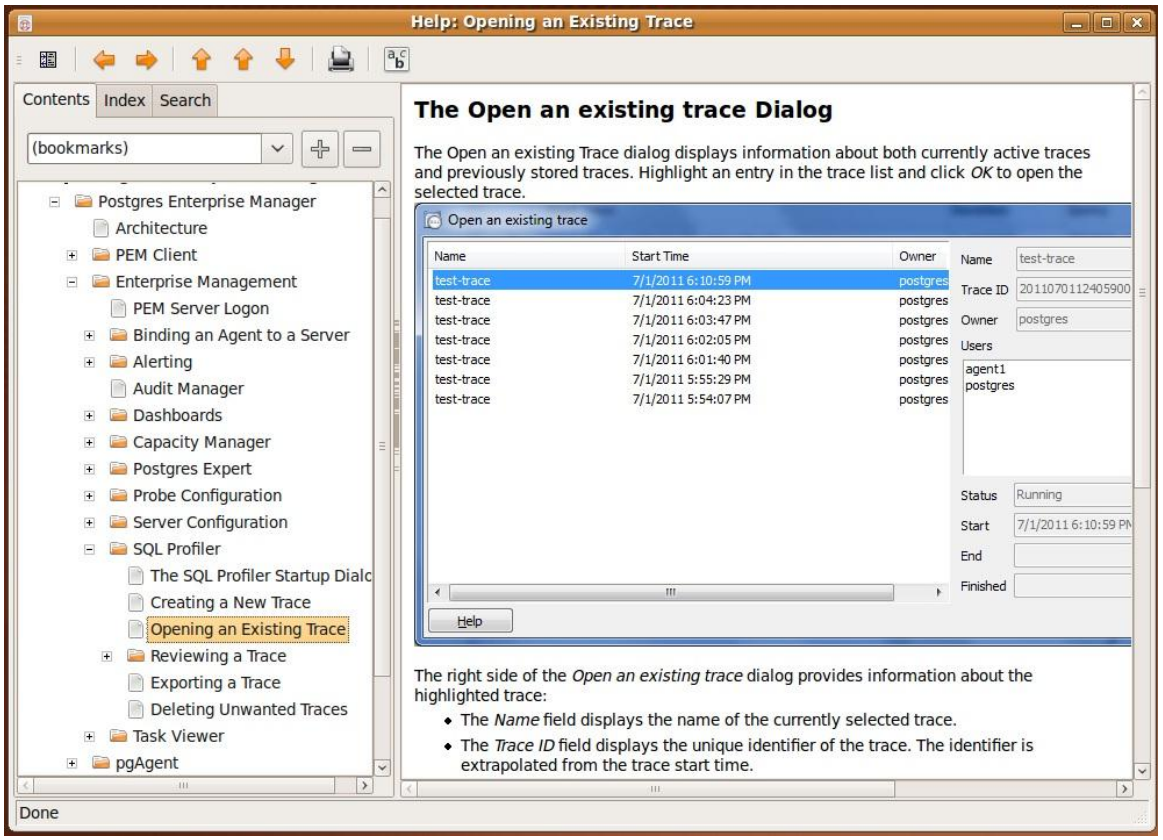

Figure 3.2 - PEM's Help dialog.

3.4 Logging on to the PEM Server

To logon to the PEM server, navigate through the `File` menu, and select the `Enterprise Manager Logon` menu option, or click the `Logon` toolbar button (as shown in Figure 3.3).

Figure 3.3 - PEM's Logon Button.

When the `PEM Server Logon` dialog opens, provide the PEM server's IP address or host name, and the user name and password specified during installation. A successful login will prompt the PEM client to display a new node in the control tree labeled `PEM Server Directory`, and a node called `PEM Agents` (PEM agents).

After logging in to the PEM server, you can add new servers to manage.

Note: Apache/PHP must be running in order for the PEM client to connect to the PEM server. On Linux, if you have trouble logging in to the PEM server, on the host where Apache/PHP is installed, check that Apache/PHP is running by entering the following command:

```
$ ps -ef | grep apache
user     13321 13267  0 16:37 pts/1    00:00:00 grep apache
```

If the result appears similar to the preceding example, Apache/PHP is not running.

Start Apache/PHP using the following commands:

```
$ su root
Password:
$ /etc/init.d/EnterpriseDBApachePhp start
```

If Apache/PHP starts properly, then you should now observe output similar to the following:

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

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

To check if Apache/PHP is running on Windows systems, open `Control Panel`, `System and Security`, `Administrative Tools`, and then `Services`. The Apache/PHP server runs as a service named `EnterpriseDB ApachePHP`.

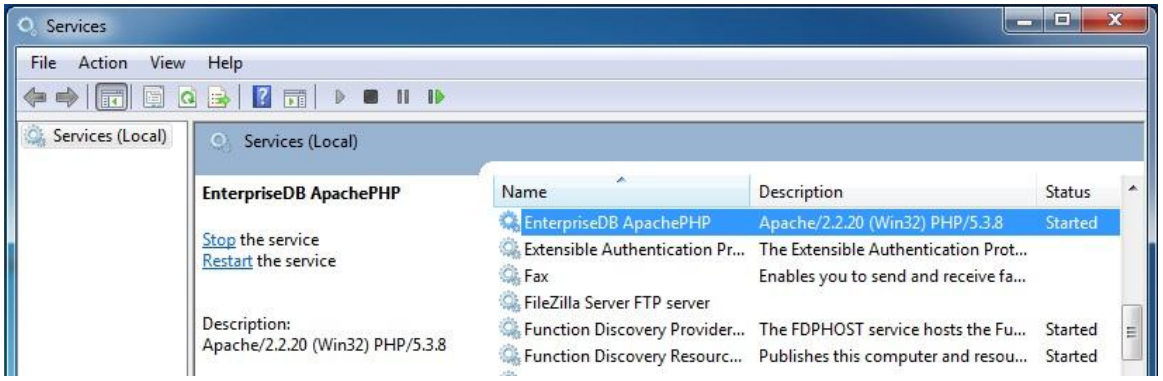

Figure 3.4 - The EnterpriseDB Apache/PHP Windows service.

Use the `Start` link to start the service if it is not running.

3.5 Adding a Managed Server

To add a new server to PEM, select the `Add Server` option from the `File` menu. You can also open the `New Server Registration` dialog by clicking the `Add Server` toolbar button.

The `New Server Registration` dialog contains four tabs on which you provide information about the new server. The first tab (`Properties`) contains general information about how to connect to the new server (see Figure 3.4)

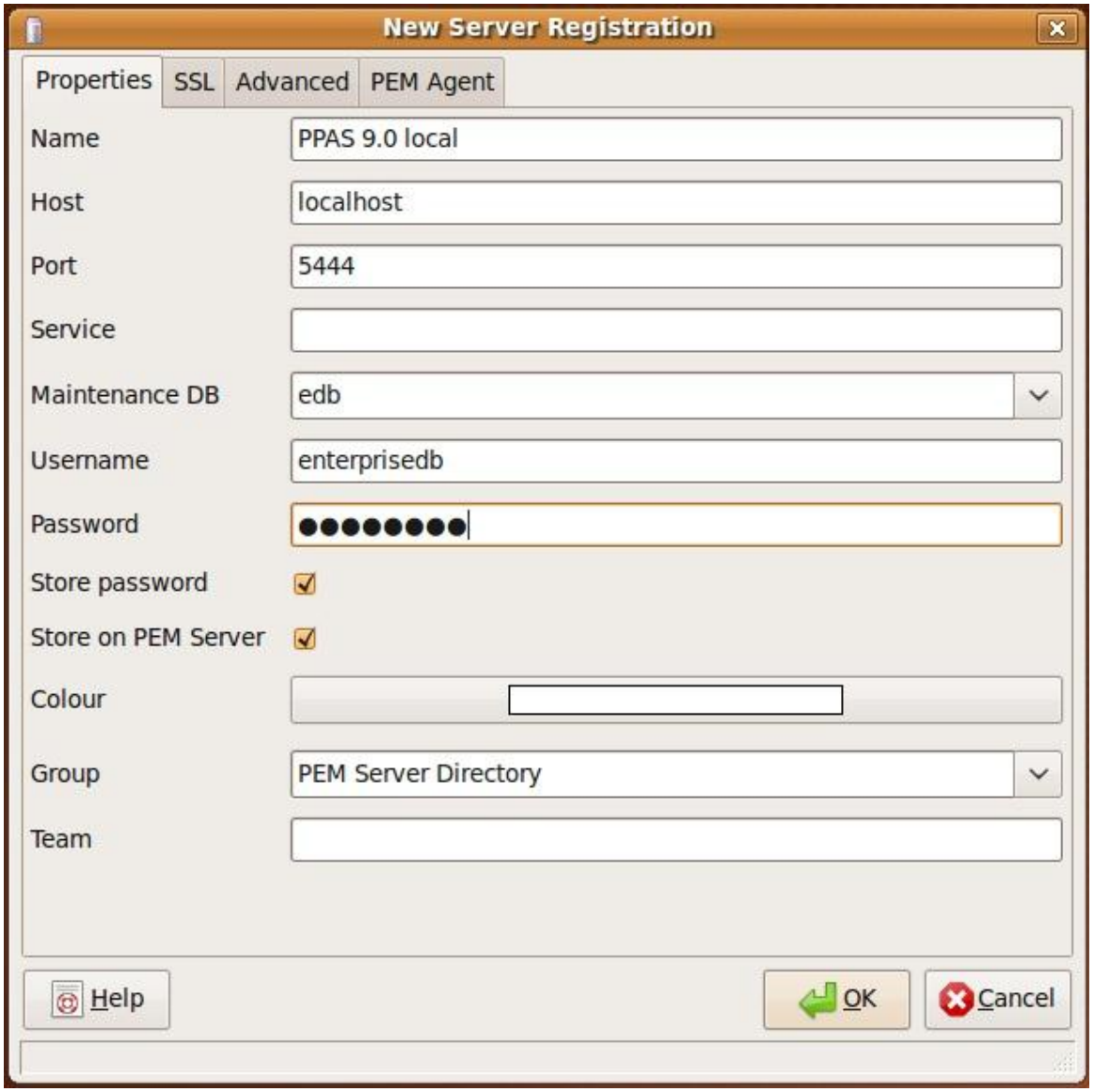

Figure 3.4 - The Server Registration dialog.

Use the fields on the Properties tab to supply information about the monitored server.

- Provide a descriptive Name for the new server.
- Specify the name or IP address of the server's Host.
- Specify the Port number that the server is monitoring.
- Specify the name of a maintenance database in the Maintenance DB field.
- Specify the Username and Password that PEM should use when connecting to the server.

To ensure that the new server will be managed by PEM, check the Store on PEM Server checkbox.

Use the fields on the SSL tab to provide authentication information specific to your system.

Use the fields on the Advanced tab to specify advanced configuration details. To allow the PEM server to remotely control startup and shutdown of the monitored Postgres server, enter the service name in the service ID control field.

Use the PEM Agent tab to specify an agent binding for the new server:

- Use the drop-down list box in the Bound Agent field to select the PEM Agent that is installed on your managed host system. Each server should be assigned a unique agent.
- Specify the host server's IP address in the Host field.

  Please note that you can specify a different Host address on the PEM agent tab than the address used on the Properties tab. For example, you can configure the PEM agent to connect via a Unix socket to the monitored server, while the client connects using an SSL connection to a forwarded port on an external firewall (that is not appropriate for the agent).
- Specify the Port number of the managed system.
- Use the SSL field to specify an SSL operational mode.
- Use the Database field to specify the name of the Postgres Plus database to which the agent will initially connect.

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

- Provide 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; using a `.pgpass` file for example.

When you've completed the `New Server Registration` dialog, click `OK`. The name of the new server should be included under the `PEM Server Directory` node in the tree control.

3.6 Remotely Starting and Stopping Monitored Servers

PEM allows you to startup and shutdown monitored server instances on remote machines from the PEM client; each remote server must be properly configured to allow the PEM client to stop or restart the server.

To remotely control the startup or shutdown of a configured server:

- Register the database server with PEM (it must appear in the tree control under the `PEM Server Directory` node).
- Right click on the server name, and select `Disconnect` from context menu.
- Then, right click on the server name and select `Connect` from the context menu. The context menu should include two new options: `Queue Server Startup` and `Queue Server Shutdown` (shown in Figure 3.5).

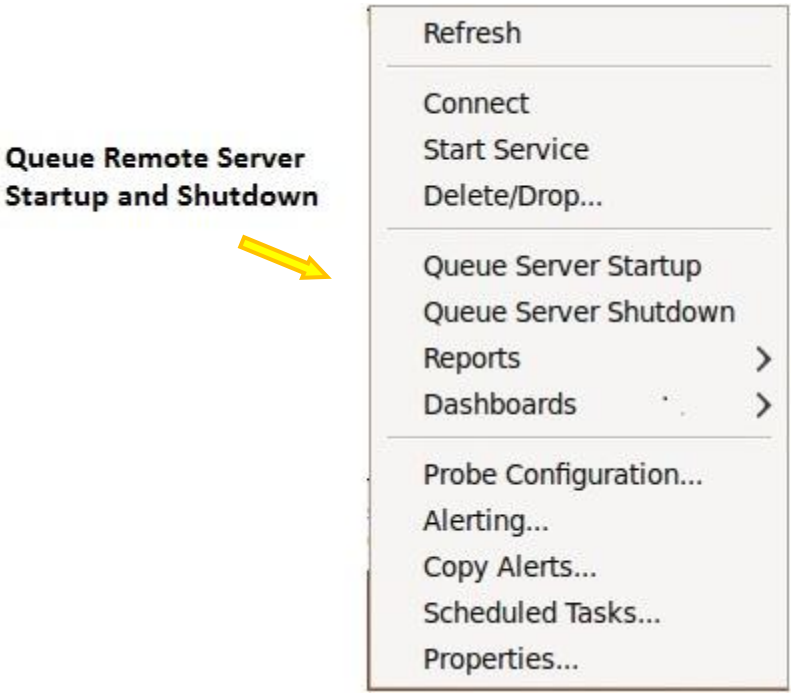

Figure 3.5 - The context menu of a remote, managed server.

Note that currently, PEM only supports the fast shutdown option of the database server.

4 General Database Administration

Postgres Enterprise Manager™ is based on the most popular GUI tool for PostgreSQL, pgAdmin. If you are already familiar with pgAdmin, you may wish to skip this section of the document; it covers some of the basic administrative tasks that can be carried out with PEM, most of which are also features in pgAdmin.

4.1 Editing a Server's Configuration

You can use the PEM client to graphically manage the configuration parameters of a remote Postgres server:

1. Connect to the managed server.

2. Use options on the `Server Configuration` menu (accessed through the `Tools` menu) to open the configuration file you wish to edit.

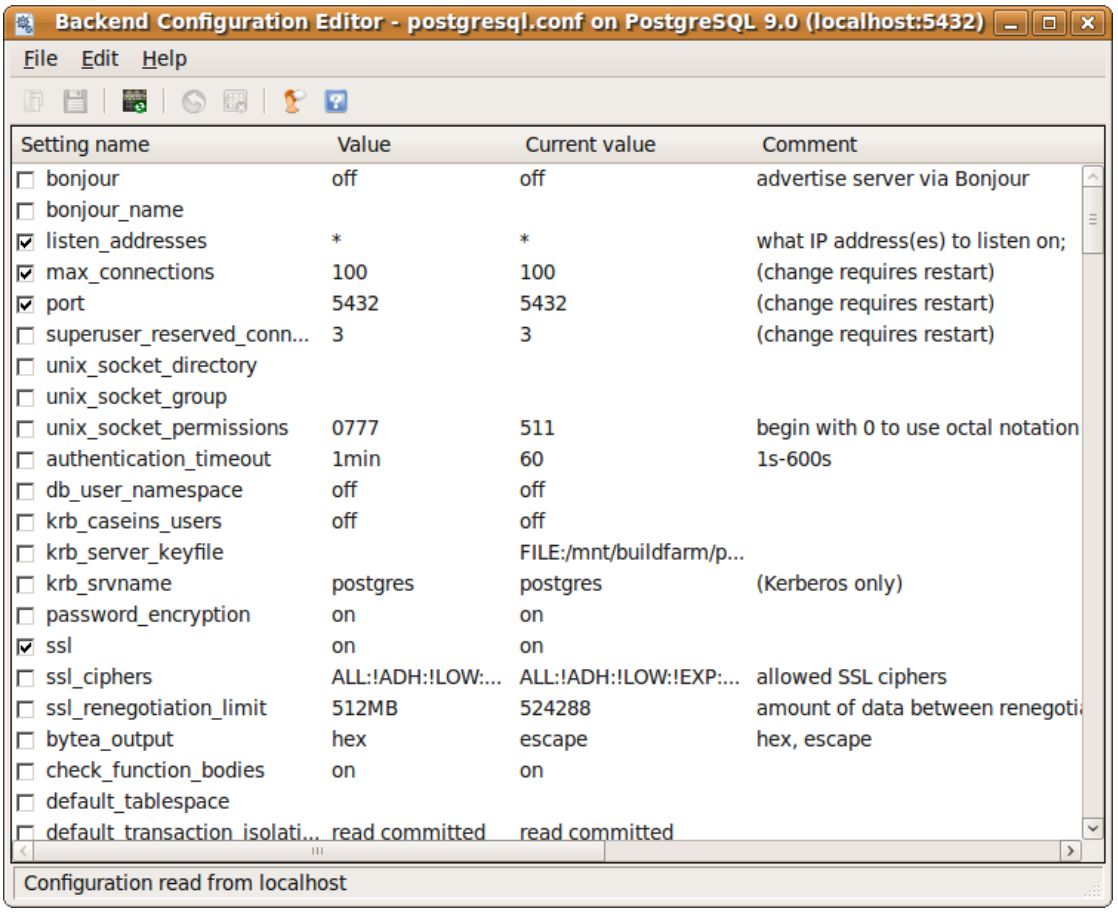

Figure 4.1 - The Configuration Editor dialog.

The Configuration Editor (shown in Figure 4.1) displays the values for the currently selected server; double-click on a parameter name to edit the parameter value. When you have made any desired changes, you can save the file on the remote server by selecting Save from the File menu, or by clicking the Save toolbar icon.

Warning: Specifying invalid values for parameters may prevent your Postgres server from starting.

When you have saved your configuration file, you must reload the server configuration. To reload the configuration files, navigate through the File menu to select Reload Server, or click the Reload Server toolbar.

Note that some parameter value changes are not dynamic and will not take effect unless the server is stopped and restarted.

4.2 Managing Security

PEM provides a graphical way to manage the security aspects of your Postgres servers. The three most common tasks are:

- The creation and maintenance of login roles
- The creation and maintenance of group roles
- Administering object permissions

4.2.1 Login Roles

A user must have a login account to connect to the Postgres server. Use the Login Role dialog (shown in Figure 4.2) to add a new login role or manage the properties of an existing login role on a registered server.

To add a new login role, right click on the Login Roles node (located beneath the selected server in the Postgres Enterprise Manager node of the tree control), and select New Login Role from the context menu.

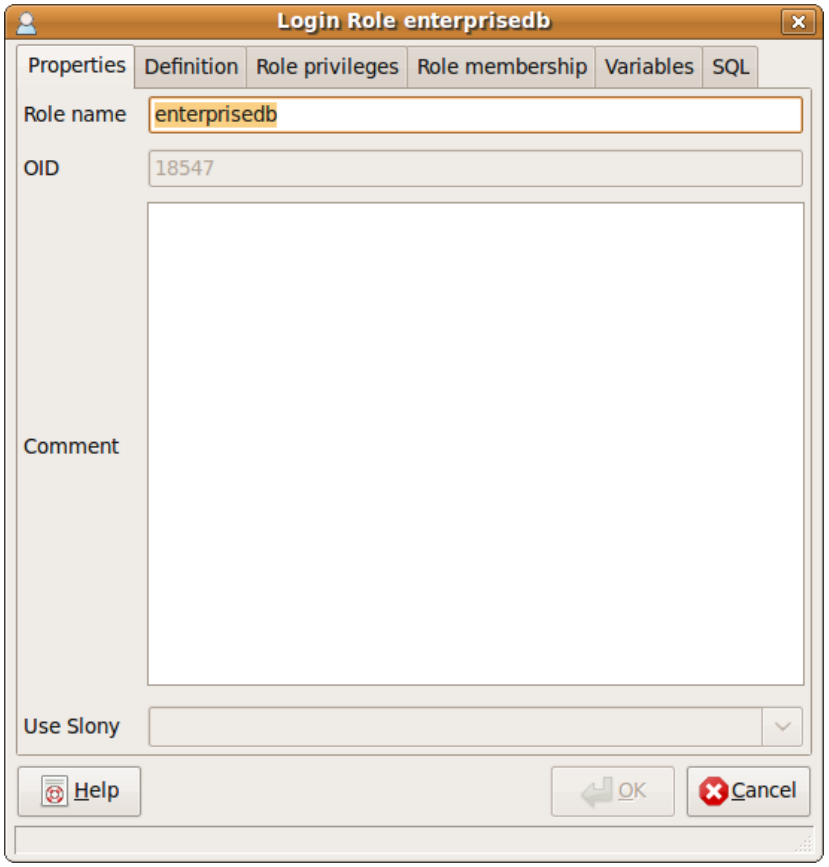

Figure 4.2 - The Login Role dialog

To modify the properties of an existing login role, right click on the login role name 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 on creating and managing login accounts, see the PostgreSQL online documentation:

http://www.postgresql.org/docs/9.0/static/sql-createuser.html

4.2.2 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 having to assign and change 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/9.0/static/sql-createrole.html

4.2.3 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.

By default, PEM displays only group roles on the Privileges tab of the Properties dialog. To instruct the PEM client to include login roles in the User/Group list on the Privileges tab, navigate through the File menu, to open the Options dialog. Select the Preferences tab; check the box next to Show users for privileges to include login roles on the Privileges tab.

The PEM client also contains a Grant Wizard (accessed through a schema node of the tree control) that allows you to manage many object permissions at once.

4.3 Managing Storage

PostgreSQL uses a named *tablespace* to define an on-disk location (a physical container) in which to store system and user data. Each PostgreSQL host may contain a single tablespace or multiple tablespaces. The PEM client provides a graphical interface that allows you to create and manage PostgreSQL tablespaces.

Use the Tablespaces node in the PEM tree control to create and manage tablespaces. The Properties editor for tables and indexes allows you to specify the tablespace in which a table or index should reside.

For more information about tablespaces, see the online documentation:

http://www.enterprisedb.com/docs/en/9.0/pg/manage-ag-tablespaces.html

4.4 Creating and Maintaining Databases and Objects

Each instance of the 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 `Database` node and select the `New Database...` menu option. You can also access the `New Database` dialog by navigating through the `Edit` menu (on the `Main` menu) to the `New Object` menu, and selecting `New Database`.

Once you have defined a database, you can create objects within the new database. Note that within each database there exist one or more *schemas*. A schema can be thought of as a directory of an operating system disk; it allows the logical separation of database objects (i.e. tables, indexes, SQL functions, and more) inside of a database.

PEM provides graphical dialogs for the creation and maintenance of all supported objects:

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

If you are using EnterpriseDB's Postgres Plus Advanced Server, you can also use the PEM client to create and manage Oracle-compatible objects (such as packages).

Each managed object is displayed in the tree control. Right click on a named node and use the context menu (or navigate through the top level menu) to perform administrative tasks for the highlighted object.

For example, to create a new table, right click on a `Table` node, select `New Table...` from the context menu. When the `New Table` dialog opens, specify the attributes of the new table (see Figure 4.3).

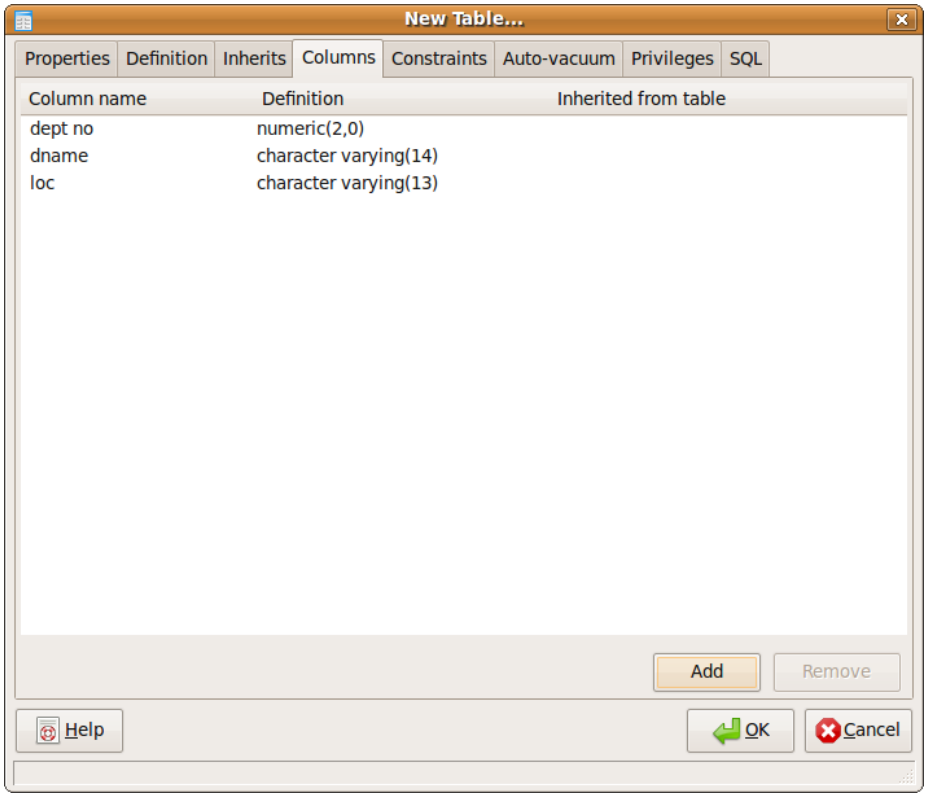

Figure 4.3 - Use PEM's dialogs to create and manage database objects.

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

4.5 SQL Development

PEM contains a feature-rich Interactive Development Environment (IDE) that allows you to issue ad-hoc SQL queries against Postgres servers.

To invoke the `Query Tool` SQL IDE from within PEM, simply highlight the name of the database you want to query in the tree control, and click the SQL toolbar icon. You can also open the `Query Tool` IDE by selecting `Query tool` from the `Tools` menu.

The `Query Tool` dialog provides an interface that allows you to manually enter in SQL queries, graphically create and execute SQL statements from dragging and dropping objects onto a visual palette, `EXPLAIN` queries and much more.

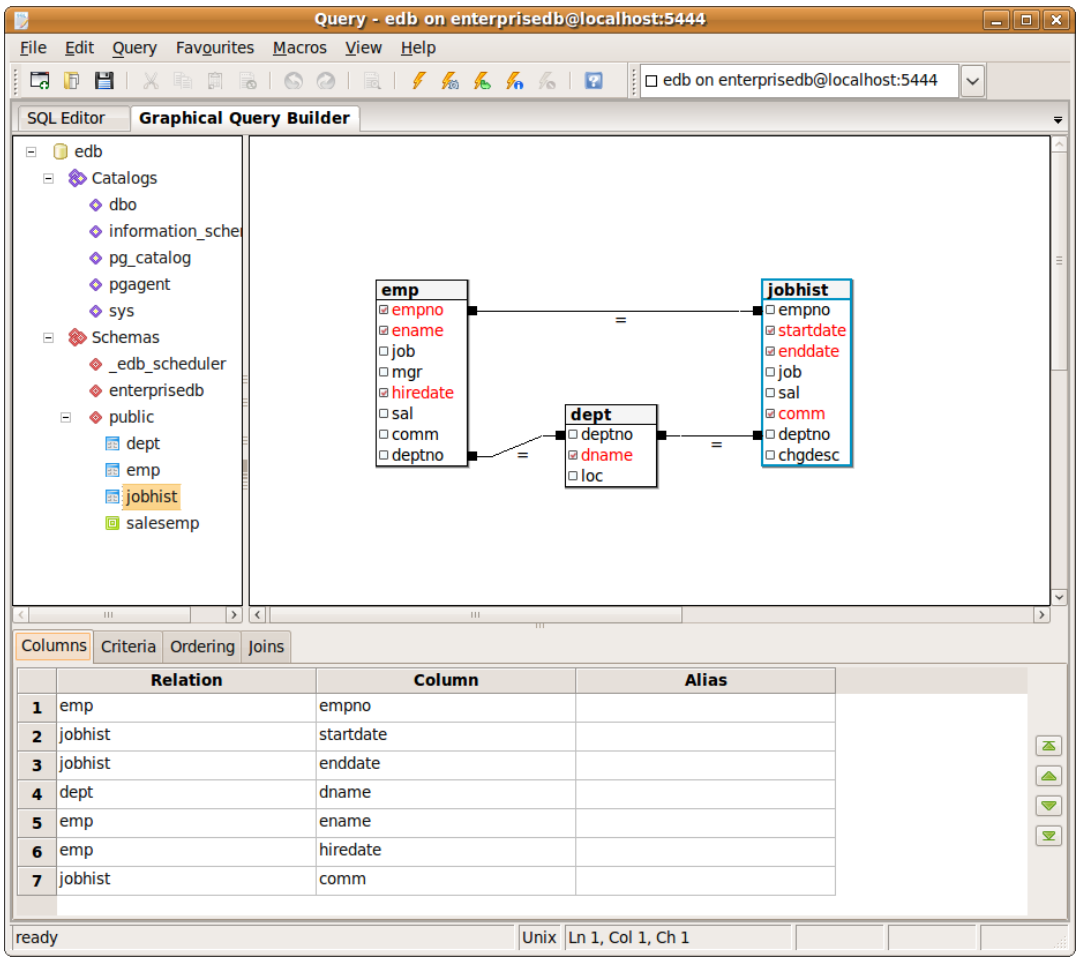

Figure 4.4 - The SQL IDE's Graphical Query Builder.

The upper part of the `Query Tool` contains the SQL Editor; you can manually enter a query, or read the query from a file. If you are manually entering a SQL query, the edit entry window also contains autocompletion code and formatting features that help you write queries.

If you prefer to build your queries graphically, you can use the Graphical Query Builder (shown in Figure 4.4) to generate SQL commands. After constructing a graphical query, click the Execute button to display the query text in the SQL Editor tab, and the results of the query in the Output pane (shown in Figure 4.5).

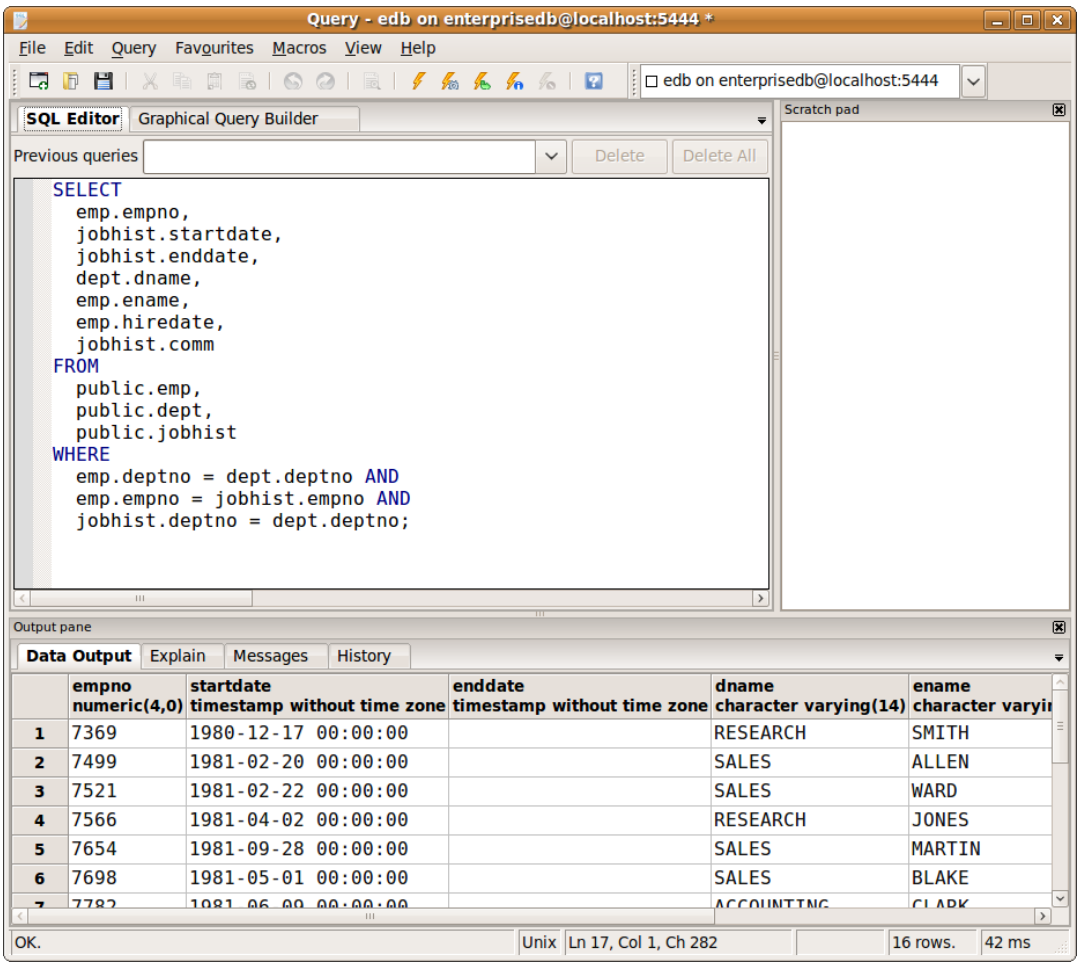

Figure 4.5 - The SQL Editor pane displays the text of the query.

You can manually adjust the query in the SQL Editor pane. As with all PEM features, online Help text is available with the click of a button.

5 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.

5.1 How to View Performance Information

PEM displays performance statistics through a number of *dashboards* that you can navigate in web browser fashion. Each dashboard contains a series of summary views that contain charts, graphs and tables that display the statistics related to the selected object.

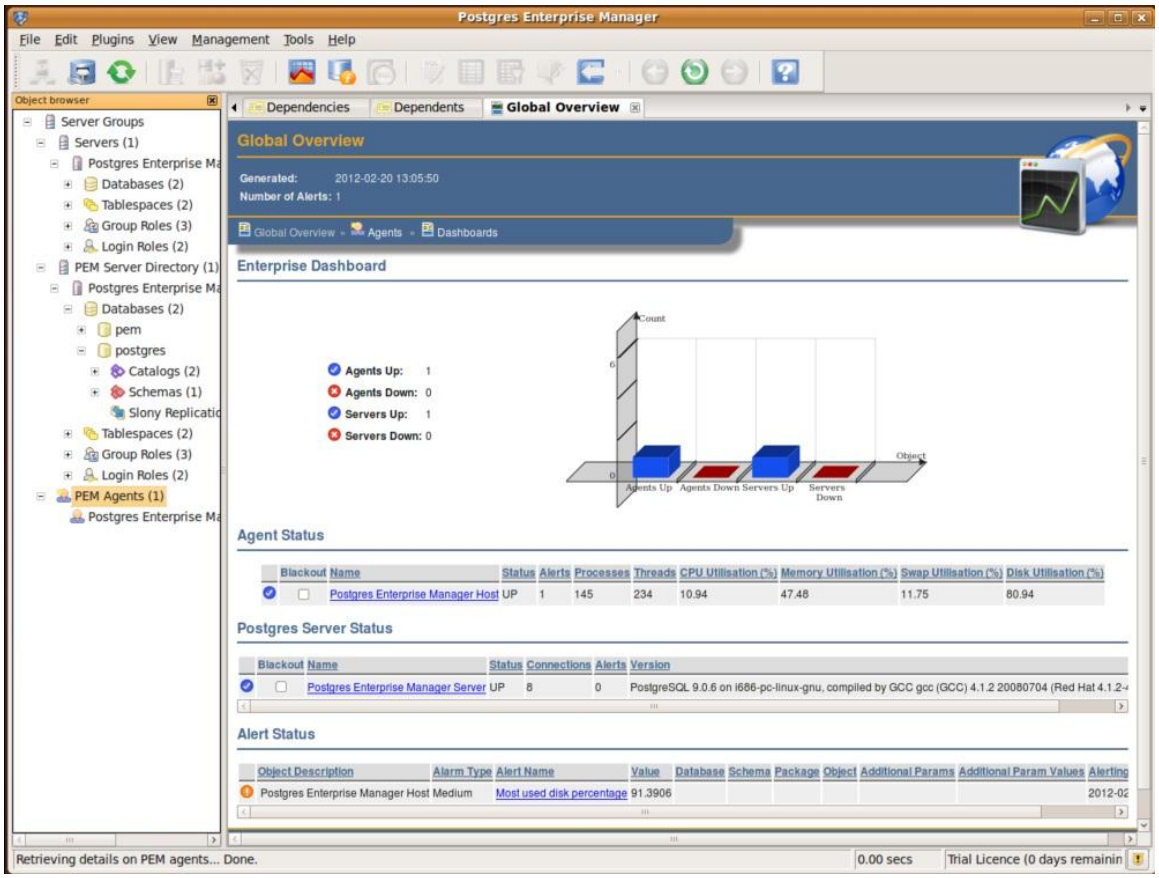

Figure 5.1 - The Global Overview dashboard.

The top-level dashboard is the `Global Overview` (shown in Figure 5.1). 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. The PEM client displays the `Global Overview` when it connects to the PEM server.

There are a number of other dashboards that provide statistical information about monitored objects. These include the:

Server Analysis dashboard

The `Server Analysis` dashboard provides general performance information about the overall operations of a selected Postgres server.

Database Analysis dashboard

The `Database Analysis` dashboard displays performance statistics for the selected database.

Memory Analysis dashboard

The `Memory Analysis` dashboard supplies statistics concerning various memory-related metrics for the Postgres server.

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.

Storage Analysis dashboard

The `Storage Analysis` dashboard displays space-related metrics for tablespaces and objects.

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.

System Waits Analysis dashboard

For Postgres Plus Advanced Server users, the `System Waits Analysis` dashboard displays a graphical analysis of views that show system wait information.

Audit Log dashboard

For Postgres Plus Advanced Server users, the `Audit Log` dashboard allows you to browse the audit logs that have been collected from instances that have audit logging and collection enabled.

To open a dashboard:

- Select the dashboard name from the `Dashboards` menu (accessed by navigating through `Management` on the `Main` 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 presented in PEM's tabbed interface (shown in Figure 5.2), opened by default in the right hand side of the client window.

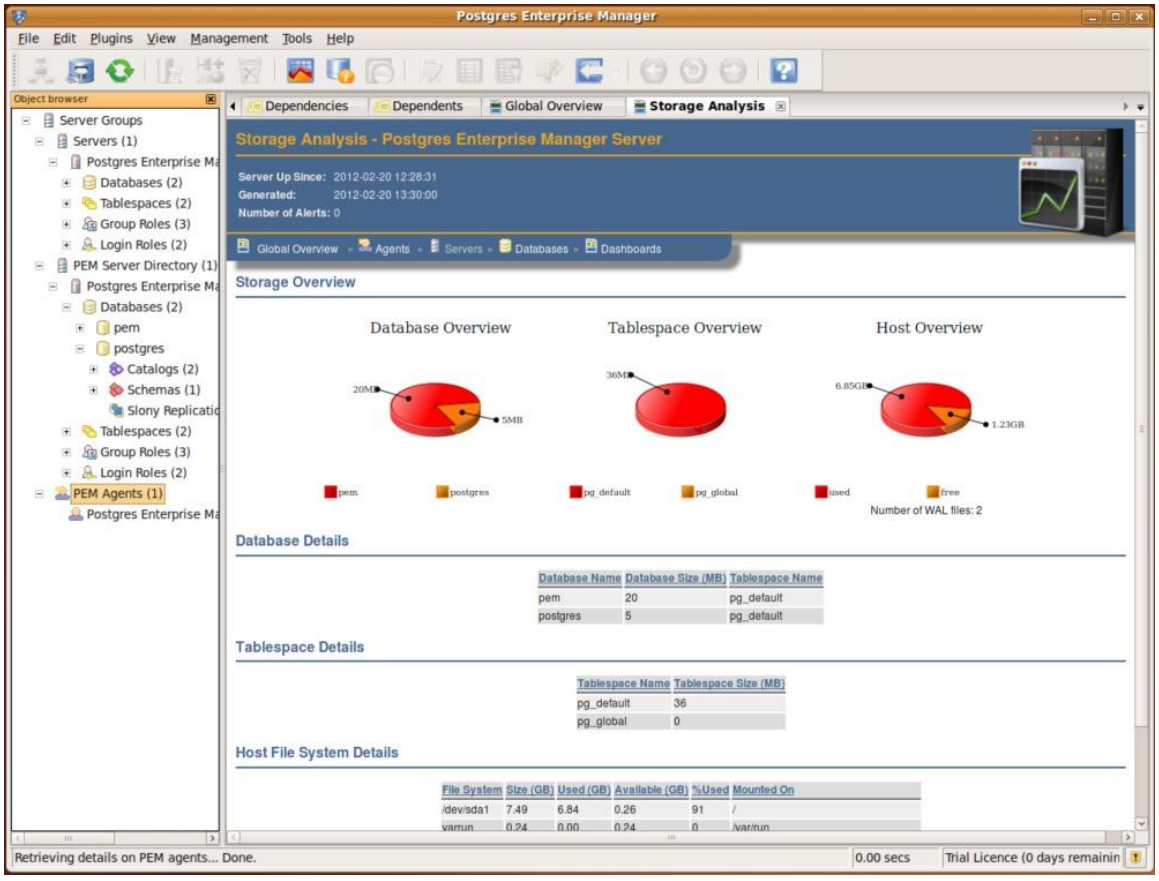

Figure 5.2 - The PEM client window, displaying a dashboard.

After opening a dashboard, you can navigate to other dashboards within the same tab.

Each dashboard header includes navigation menus that allow you to open dashboards that contain statistical information for the currently selected object, and any object that resides beneath the selected object (in the tree control hierarchy). You can scroll backward and forward through the dashboards that you've opened within a tab by using the `Browser Back` and `Browser Forward` buttons (on the main toolbar). You can update the current dashboard by clicking on the `Refresh` button.

Click the `X` on the dashboard tab to close a dashboard.

5.2 Viewing and Responding to Alerts

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*. Alerts call your attention to conditions on registered servers that require your attention.

You can view alert information in a number of places inside PEM:

- The `Global Overview` dashboard contains a count of all alerts that have occurred, along with a breakdown of total alerts by monitored server and a listing of the most recent alerts that have occurred.

- The `Alerts Overview` dashboard displays summarized statistics for all alerts that have occurred across your servers as well as a detailed listing of each alert that has been identified.

- When an alert is triggered, a flashing  icon is displayed in the lower right-hand corner of the main window. Click the icon to open the `Alerts Overview` dashboard.

To open the `Alerts Overview` dashboard, right click on a server or agent node, and select `Alerts Dashboard` from the `Dashboards` menu. You can also open the `Alerts Dashboard` by navigating through the `Dashboards` menu (located on the `Management` menu).

5.3 Customizing Probes and Alerts

PEM uses *probes* to retrieve statistics from a monitored server, database, operating system or agent. A probe is a scheduled event that returns a set of performance metrics about a specific monitored object. You can use the `Probe Configuration` dialog (shown in Figure 5.3) to specify when each probe is executed.

To review or modify the currently defined probes for each server and its underlying objects (e.g. databases, tables, etc.), right click on the object name in the tree control and select `Probe Configuration` from the context menu. You can also access the `Probe Configuration` dialog by highlighting an object name, and selecting `Probe Configuration` from the `Management` menu.

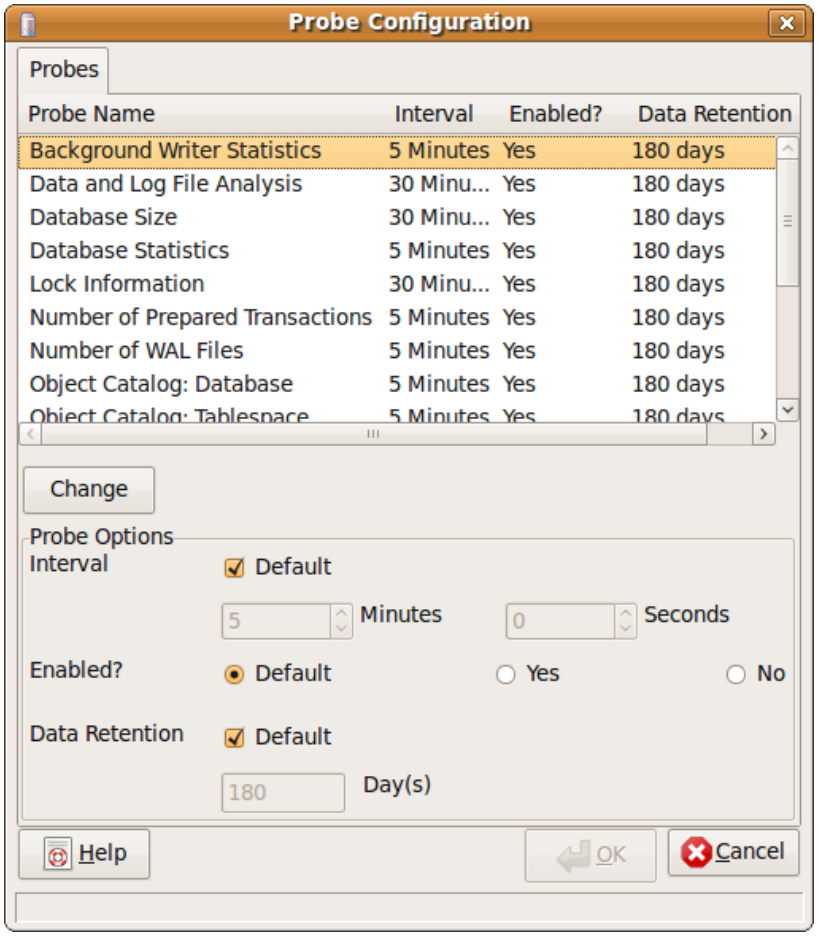

Figure 5.3 - The Probe Configuration dialog.

Probes monitor a unique set of metrics for each specific object type (server, database, database object, or agent). You can modify the properties associated with a probe, specifying:

- How often the probe executes.
- How long its information is retained for historical reporting purposes.
- If the probe is enabled or disabled.

The `Probe Configuration` dialog displays a list of the metrics that are collected for the selected node. The `Probe Configuration` dialog may also display information about probes that cannot be modified from the current node. If a probe cannot be modified from the current dialog, the background (the area behind the node name) is greyed-out; when highlighted, the status bar will display the node level from which the statistic may be configured.

PEM provides very granular control over your alerts, allowing you to define alerting thresholds for monitored objects that meet the requirements of your system.

Use the `Alerting` dialog (shown in Figure 5.4) to define or modify an alert. To open the `Alerting` dialog, right click on the name of a monitored object in the tree control, and select `Alerting...` from the context menu.

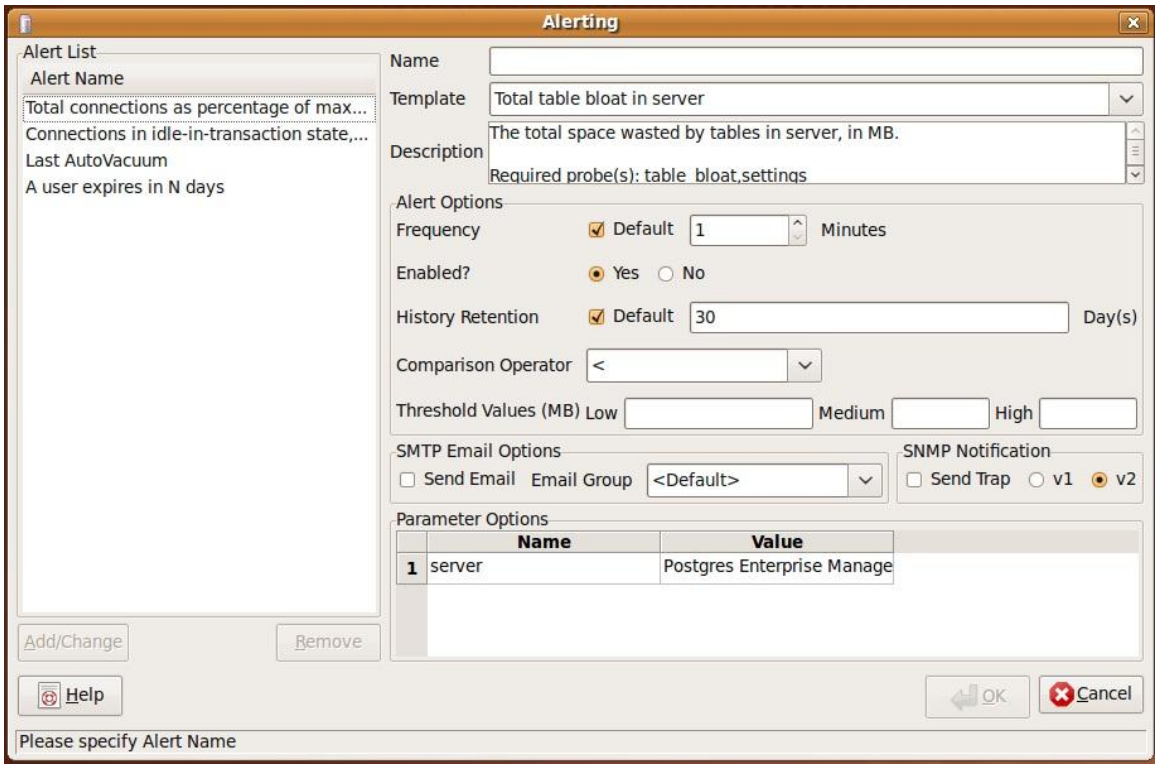

Figure 5.4 - The PEM Alerting dialog

To define an alert, provide a user-friendly name, and select a monitoring statistic that applies to the selected object (e.g. database, table, etc.). Use the fields in the `Alert Options` portion of the `Alerting` dialog to specify properties for the alert:

- How often PEM will test the alert conditions
- If the alert is enabled (or disabled)
- How long information gathered is retained

Then, specify the threshold conditions and parameter options that will trigger the alert.

You can also configure SMTP email notifications or SNMP notifications for the alert.

When you've defined an alert, click the `Add/Change` button to save the alert; when you've defined or modified all of the alerts for a specific node, click the `Ok` button to make the changes persistent, and to instruct PEM to begin enforcing the alerts.

6 Capacity Planning

PEM contains built-in capabilities for performing database capacity planning. Capacity planning helps DBAs by providing answers to questions like:

- How much storage will my database need six months from now?
- How fast is my database growing?
- What objects are responsible for the growth in my database?
- Will my server be able to support another database instance?
- Is the performance of my database getting better, staying the same, or getting worse?

Capacity planning for databases typically involves two things:

- Historical trend analysis, which involves viewing selected database statistics over various time periods so that trends can easily be spotted.
- Forecasting, which entails using historical statistical information and projecting the values of various statistics (e.g. a database's size) will be in the future.

6.1 Performing Trend Analysis

PEM automatically collects a wide range of performance metrics about storage usage, memory usage, I/O traffic and more. The performance metrics are stored in a metadata repository that is created when PEM is installed.

Of course, you have full control over what and how often data is collected, but you can also take advantage of the product's defaults and have the recommended statistics gathered for you automatically.

The hard part of capacity planning operations (automatic data collection) is transparently handled for you by PEM.

All that remains is to use PEM's Capacity Manager interface to build desired trend analysis and forecasting reports.

To open the `Capacity Manager` dialog, select `Capacity Manager...` from the `Management` menu. The `Capacity Manager` dialog (shown in Figure 6.1) provides

quick access to a list of the available metrics, and the options for producing capacity planning reports.

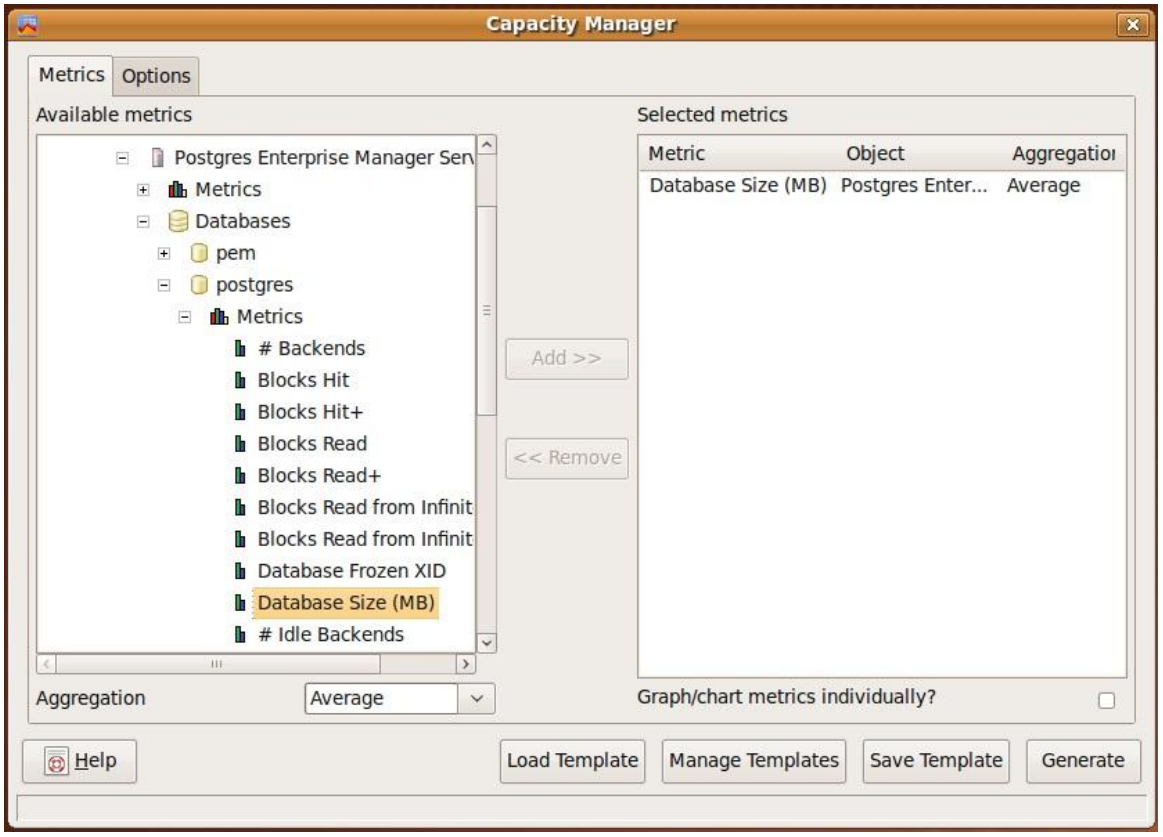

Figure 6.1 - The Capacity Manager dialog

The `Capacity Manager` dialog displays two tabs – `Metrics` and `Options` – that you can use to define capacity planning reports.

The `Metrics` tab displays a tree control that allows you to easily navigate all of your managed servers and select statistics that you wish to analyze. For example, to follow the growth of a particular database, you would expand that node in the tree control, highlight `Database Size` in the listed `Metrics`, and click the `Add >>` button to add it to the `Selected metrics` pane.

Capacity Manager can plot multiple statistics on one graph or produce a separate graph for each distinct metric. A checkbox located in the lower right corner of the `Metrics` tab of the `Capacity Manager` dialog enforces this option.

When you have specified the metrics that will be included in the analysis, you can then specify the timeframe over which the analysis will be performed. Use the fields on the `Options` tab of the `Capacity Manager` dialog, to specify the `Time Period` covered by the report (see Figure 6.2).

Capacity Manager allows you to create both graphical and tabular reports for historical trend analysis and future forecasts analysis reports. The `Options` tab provides fields that allow you to specify the form that the resulting report will take:

- A graph
- A data table
- Both a graph and data table

Finally, you can specify where Capacity Manager reports are displayed or written (see Figure 6.2). PEM will display the report in either a new or existing tab within the PEM client or write the report to a file on the host of your client workstation.

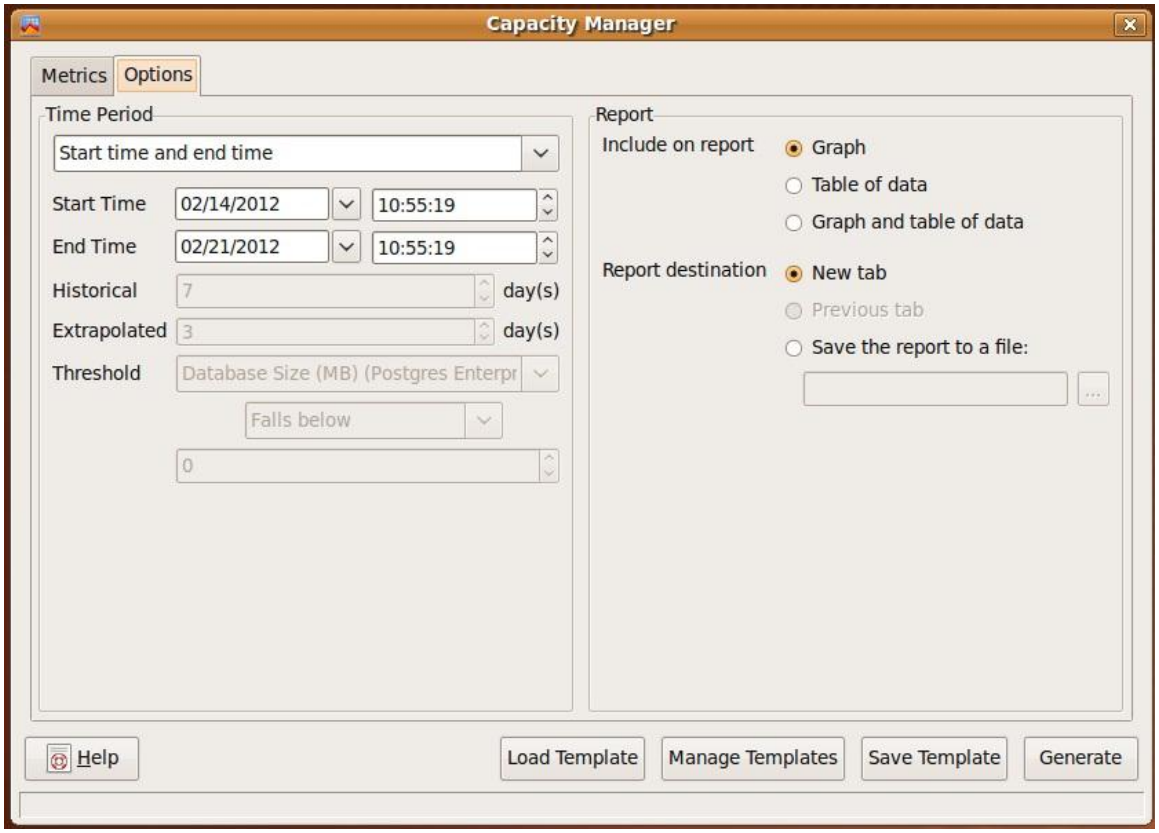

Figure 6.2 – Specify the time period, type, and destination of the Capacity Manager report.

Specify a `Report destination`, and click the `Generate` button to generate the report, as shown in Figure 6.3.

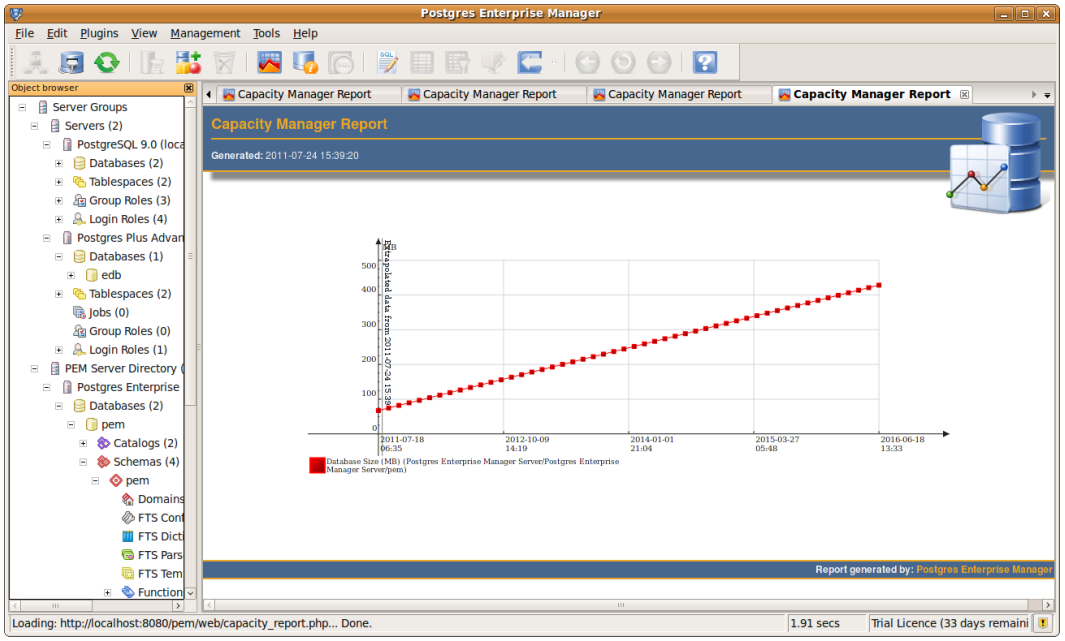

Figure 6.3 - The Capacity Manager report

6.2 Forecasting Future Trends

Capacity Manager uses historical metrics to forecast future trends. To create forecasting reports with Capacity Manager, simply select your desired metrics, and use the Capacity Manager `Options` tab to specify an `End` time for the report that is in the future (see Figure 6.4).

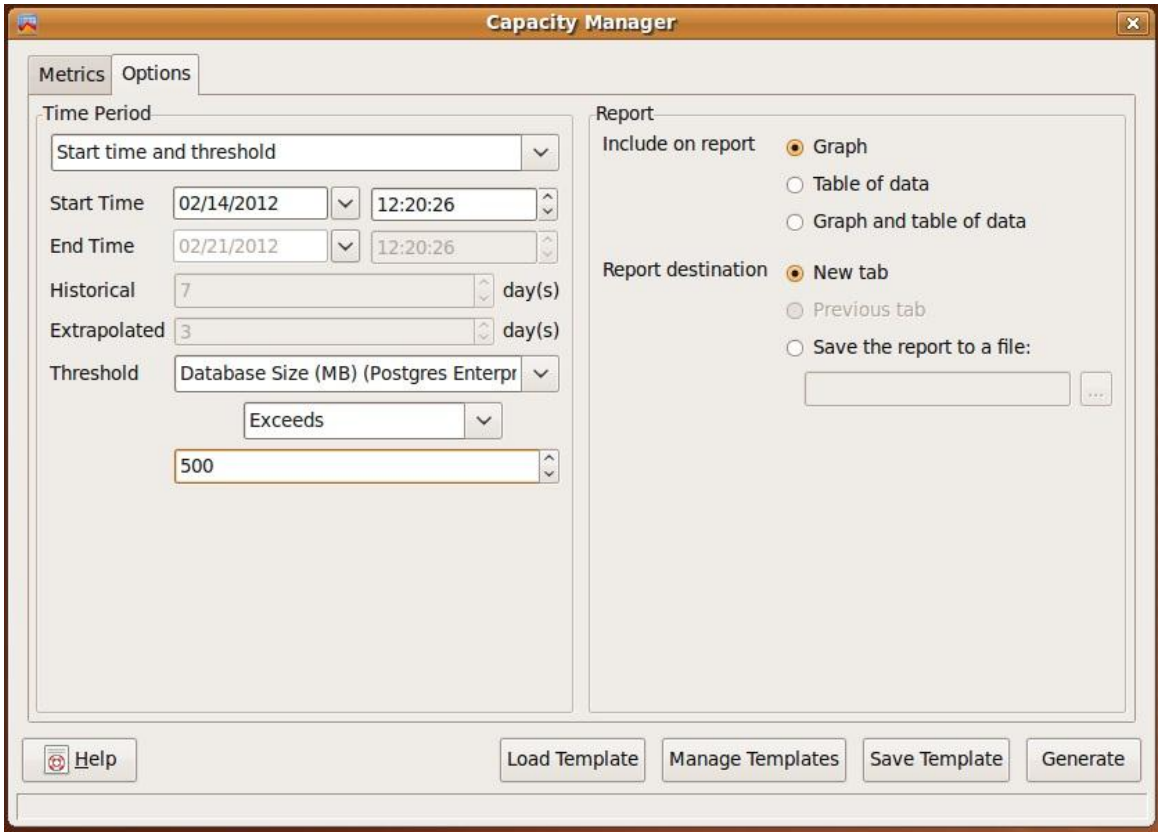

Figure 6.4 - Using the Options dialog to forecast future trends.

For example, you might use Capacity Manager to predict when you will need to increase the database storage available on your system. Select `Start time and threshold` or `Historical days and threshold` from the drop-down list box depending upon whether you want to use a specific past date/time or a number of historical days from which to draw the extrapolation data. In this example, a start date/time was chosen. Select the `Database Size (MB)` metric; use the drop-down list boxes to specify that the report should tell you when the threshold exceeds 500 Megabytes in storage. When you click the `Generate` button, Capacity Manager will use historical usage data to predict the point in time that your `Database Size` will exceed 500 Megabytes in size.

7 Audit Logging

Audit logging of activity in Postgres Plus Advanced Server instances can be done with PEM. The Audit Manager is used to configure audit logging including such attributes as the type of database activities to be recorded, when and how often audit logs are to be rotated, and how often audit logs are to be gathered into PEM.

The specific types of activities that may be chosen for recording in the audit logs are the following:

- All connections made to the database instance or only failed connections
- 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 available to PEM, the `Audit Log` dashboard is used to view the information in an easy-to-read form. The `Audit Log` dashboard provides the capability to filter the activity to be viewed by such characteristics as a timestamp range of when an activity occurred, the database on which the activity occurred, the user performing the activity, or the type of command being acted upon.

7.1 Setting the Advanced Server Instance Service ID

Before configuring audit logging on a Postgres Plus Advanced Server instance, it must be configured as a managed server, and the `Service ID` property of that Advanced Server instance must be completed with the Advanced Server service name.

Right click on the server name and select `Properties` from the context menu. Click the `Advanced` tab to verify that the `Service ID` field has been completed as shown in Figure 7.1. This allows the PEM server to stop and start the service.

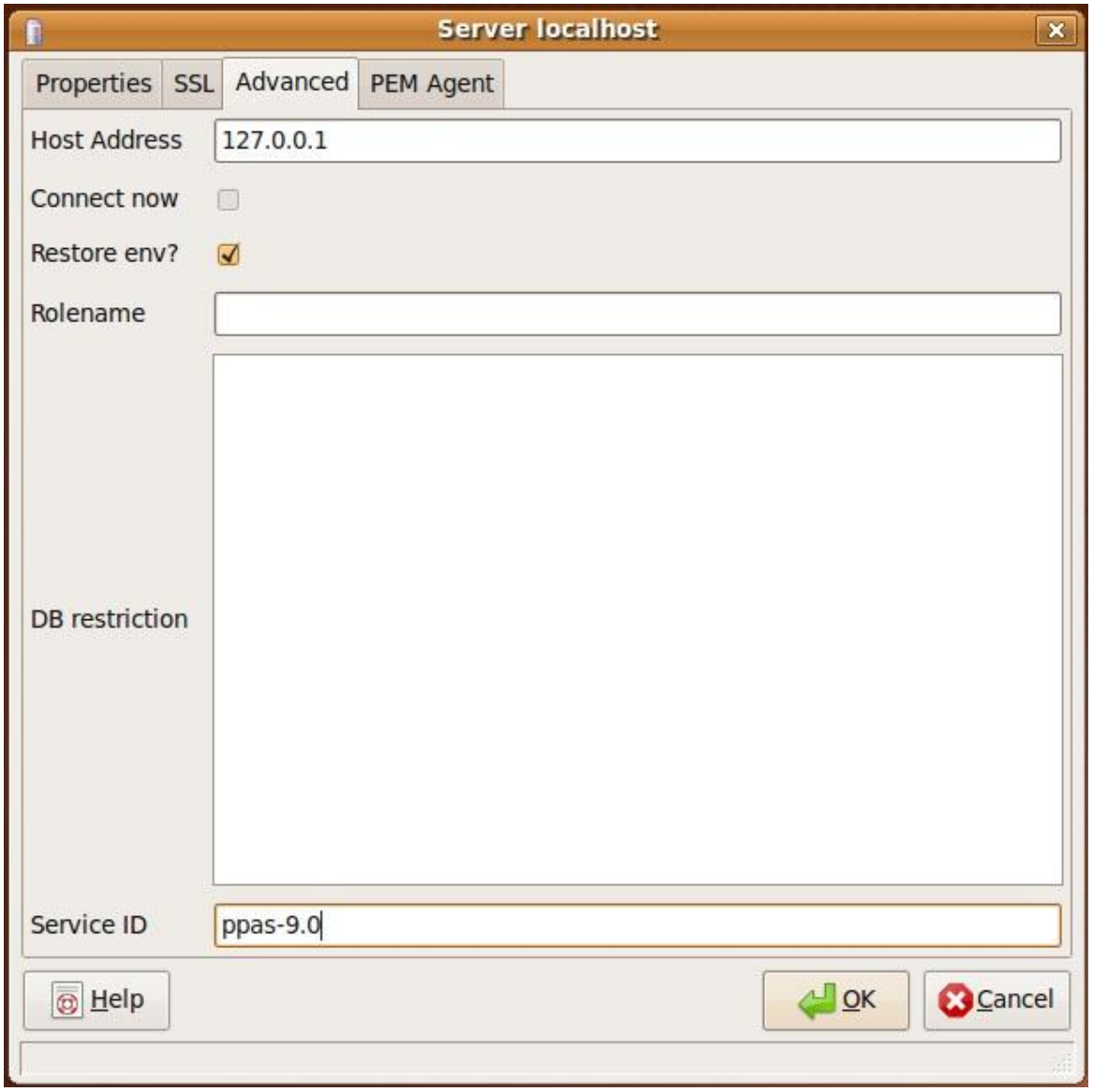

Figure 7.1 - The Service ID of the Advanced Server instance.

On Linux systems, provide the name of the init script located in `/etc/init.d`.

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

The EnterpriseDB installer for Postgres Plus Advanced Server uses a default service ID of `ppas-X.Y` where `X.Y` specifies the major version number of the server. For example: `ppas-9.0`.

Note that for a PostgreSQL database server, the EnterpriseDB installer uses a service ID of `postgresql-X.Y`. For example, `postgresql-9.0`.

7.2 Setting the EDB Audit Configuration Probe

The EDB Audit Configuration probe must be enabled to monitor the audit logging configuration of Postgres Plus Advanced Servers.

Right click on the Advanced Server name in the tree control and select `Probe Configuration` from the context menu. You can also access the `Probe Configuration` dialog by highlighting the Advanced Server name, and selecting `Probe Configuration` from the `Management` menu.

The `Enabled` column in the `Probe Configuration` dialog should be set to `Yes` for the EDB Audit Configuration probe (see Figure 7.2).

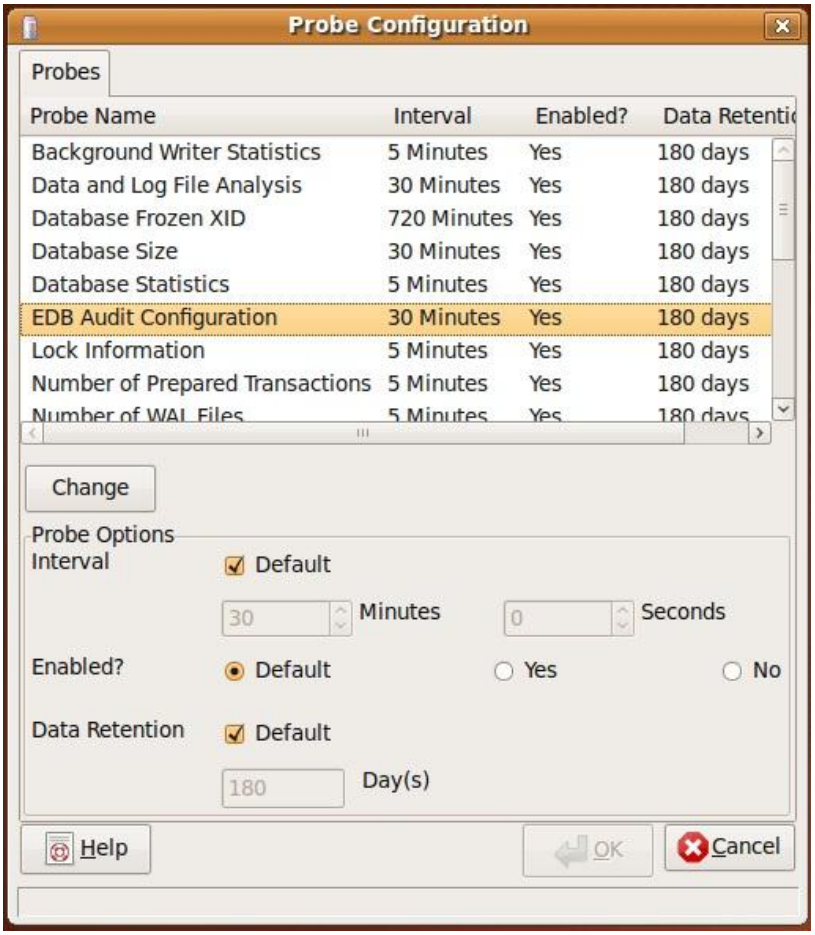

Figure 7.2 - The EDB Audit Configuration probe.

If it is not enabled, you can enable it using the `Enabled` radio buttons in the `Probe Options` dialog. You can alter other attributes such as the probe interval or data retention if desired.

7.3 Configuring Audit Logging with the Audit Manager

To open the `Audit Manager`, select `Audit Manager...` from the `Management` menu. The `Audit Manager` dialog (shown in Figure 7.3) appears. Click `Next` to continue.

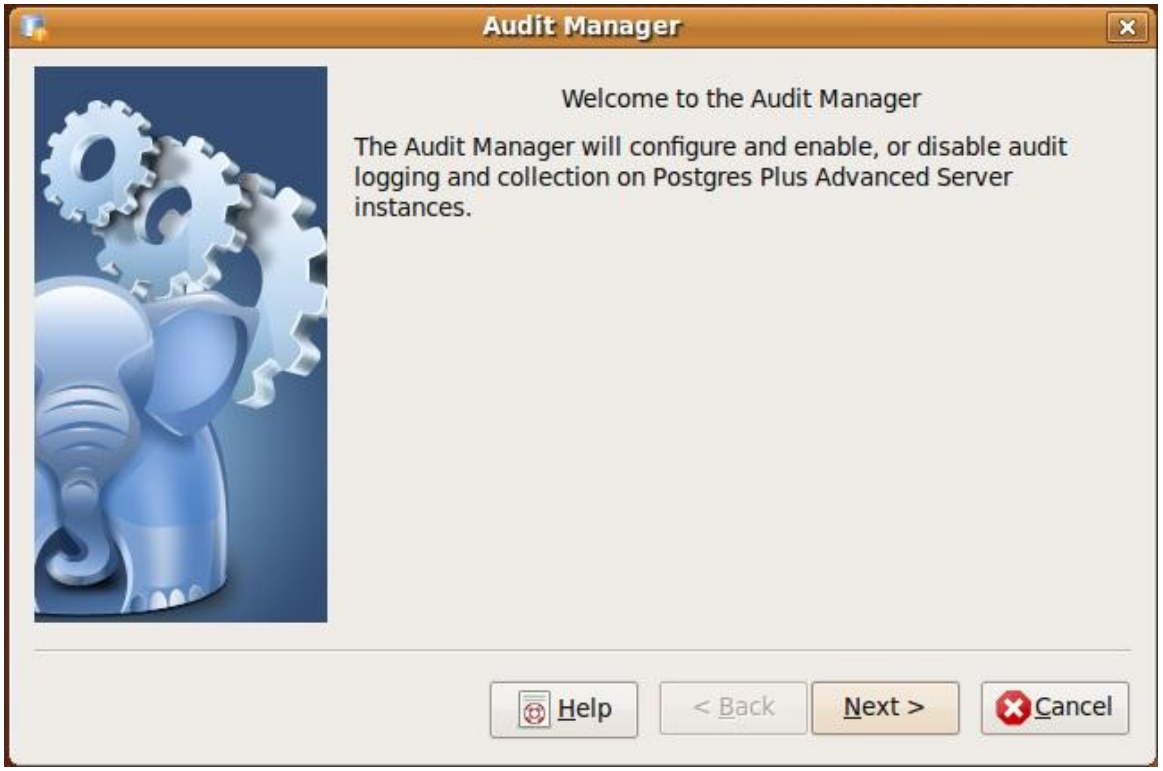

Figure 7.3 - The Audit Manager Welcome dialog.

Select the servers you wish to configure for auditing (shown in Figure 7.4). Click `Next` to continue.

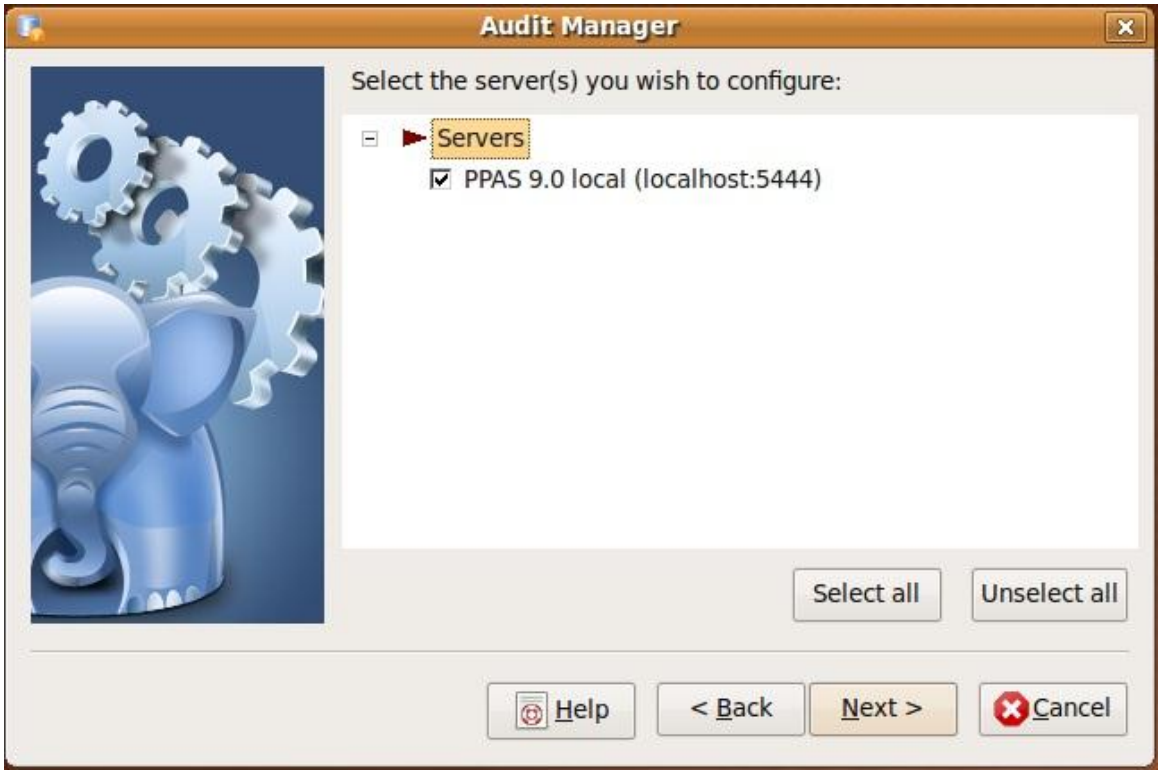

Figure 7.4 – Select the servers to configure for auditing.

The `Auditing Parameters Configuration` dialog lets you enable or disable auditing and choose how often log records are collected into PEM (see Figure 7.5).

Figure 7.5 – The Auditing Parameters Configuration dialog.

Set the options in the `Auditing Parameters Configuration` dialog as follows:

- Specify `Enable` for the `Auditing Status` to allow auditing to occur.
- Check the `Enable Log Collection` checkbox so PEM will periodically gather the log records so you can later view them in the `Audit Log` dashboard.
- Use the `Collection Frequency` drop-down list to select how often PEM is to collect the log records.

Click `Next` to continue to the `Audit Log Configuration` dialog (see Figure 7.6).

Figure 7.6 – The Audit Log Configuration dialog.

Use the `Audit Log Configuration` dialog to determine the types of activities to be logged during auditing.

- Specify `All` to log all connection attempts, `Failed` to log only failed connection attempts, or `None` for no connection logging of `Log Connection Attempts`.
- Specify `All` to log all disconnection attempts or `None` for no disconnection logging of `Log Disconnection Attempts`.
- Check the `Log Select Statements` checkbox if you want to log `SELECT` statements.
- Check the `Log Error Statements` checkbox if you want to log SQL statements that result in an error.
- Check the `Log DML Statements` checkbox if you want to log data manipulation language SQL statements such as `INSERT`, `UPDATE`, and `DELETE`.
- Check the `Log DDL Statements` checkbox if you want to log data definition language SQL statements such as `CREATE`, `DROP`, and `ALTER`.

Click `Next` to continue to the `Auditing Parameters Log Rotation` dialog (see Figure 7.7).

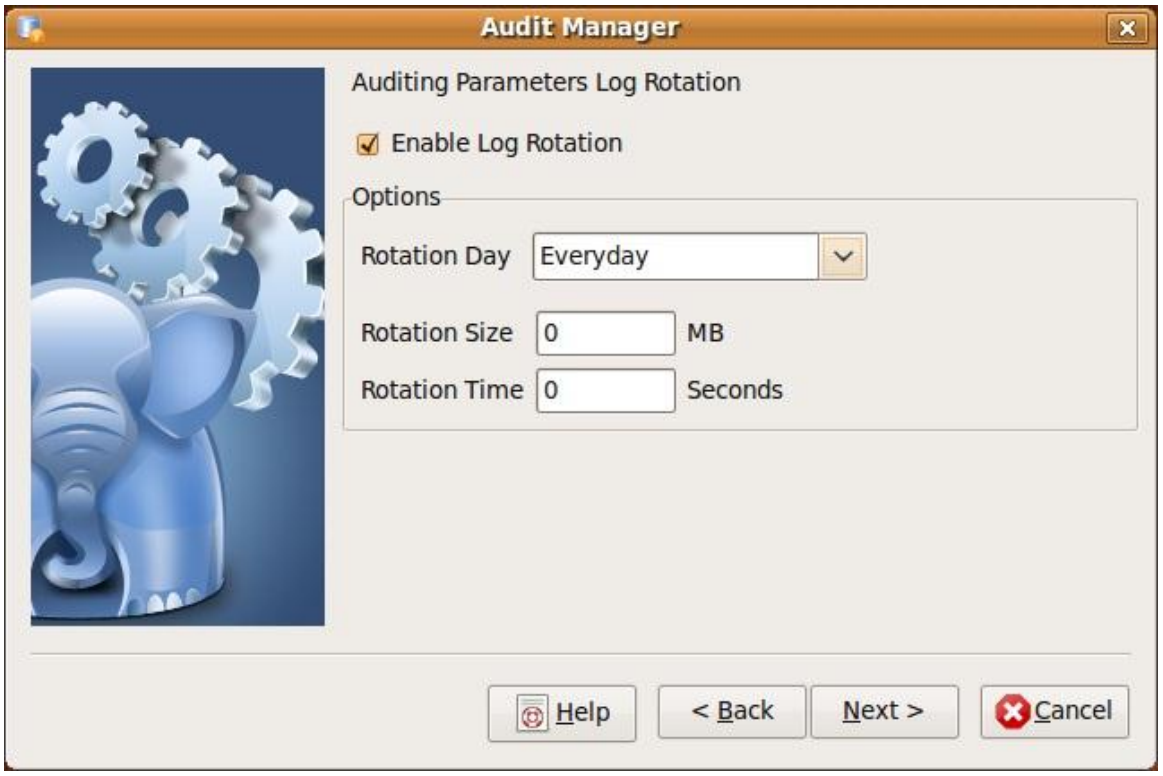

Figure 7.7 – The Auditing Parameters Log Rotation dialog.

Use the `Auditing Parameters Log Rotation` dialog to set factors controlling audit log file rotation.

- Check the `Enable Log Rotation` checkbox if you want the log file to be periodically rotated. If the log file is not rotated, a single file is used, which may grow to an unmanageably large size over time. Thus, it is suggested to enable log rotation.

- Use the `Rotation Day` drop-down list to specify the day of the week on which the log file should be rotated. Select `Everyday` if you want log rotation to occur on every day of the week. Select `None` if you do not want the log rotation criteria based on the day of the week. Otherwise, choose the desired day of the week for log rotation from the drop-down list.

- Use the `Rotation Size` field if you want to base the rotation criteria on the size of the log file. Specify the size in megabytes for when the log file should be rotated upon reaching the given file size.

- Use the `Rotation Time` field if you want to base the rotation criteria on time between rotations. Specify the number of seconds after which each rotation should occur.

Click `Next` to continue to the `Schedule Auditing Changes` dialog (see Figure 7.8).

Figure 7.8 – The Schedule Auditing Changes dialog.

Use the `Schedule Auditing Changes` dialog to determine when auditing configuration changes are to take effect.

- Select `Configure Auditing 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.

- Select `Schedule it for some other time` if you want 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 configure a scheduled job to apply the configuration to each server. The job created 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.

The scheduled jobs can be viewed in the Task Viewer, and the results in the Log Viewer when opened from the appropriate server or agent. (Right click on a server or agent and choose `Scheduled Tasks` from the menu in order to open the Task Viewer.)

7.4 Viewing the Log with the Audit Log Dashboard

Use the `Audit Log` dashboard to view the audit log from Postgres Plus Advanced Server database instances (see Figure 7.9).

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).

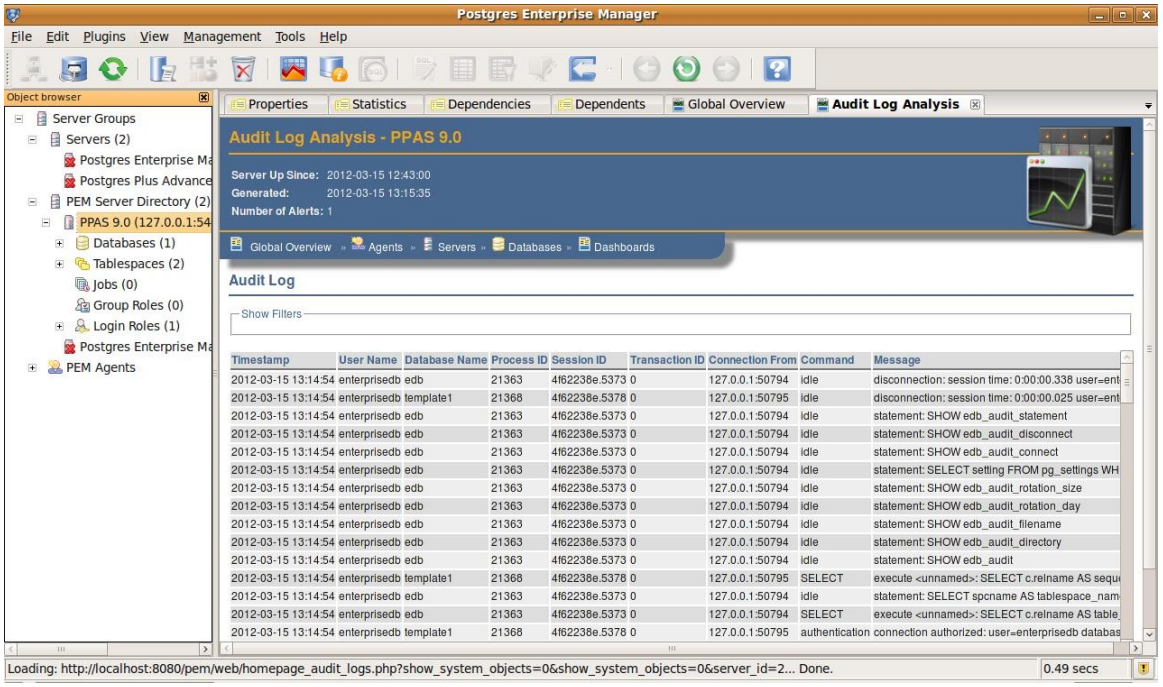

Figure 7.9 – The Audit Log dashboard.

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 (see Figure 7.10).

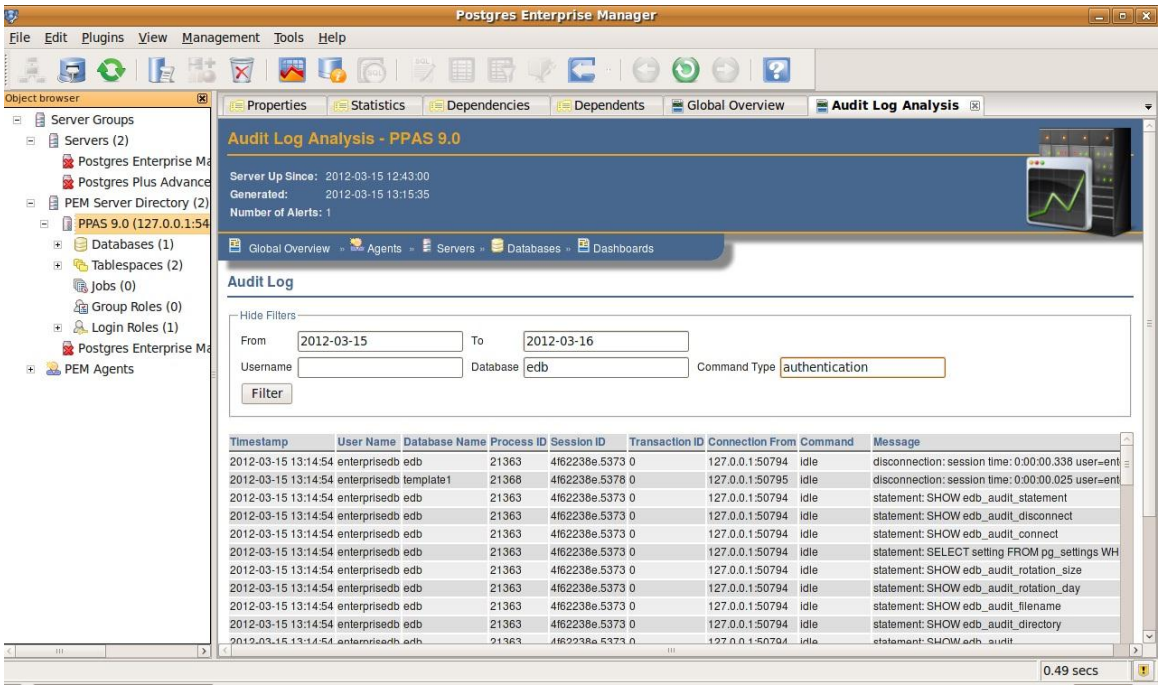

Figure 7.10 – The Audit Log dashboard filters panel.

Use the fields in the filters panel to provide certain selection criteria for the audit records you wish to display.

- Use the `From` field to specify a start date to show only log entries on or after the given date. Clicking the mouse button in the field opens a calendar from which you can make your start date selection.

- Use the `To` field to specify an end date to show only log entries made prior to the given date. Clicking the mouse button in the field opens a calendar from which you can make your end date selection.

- Use the `Username` 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.

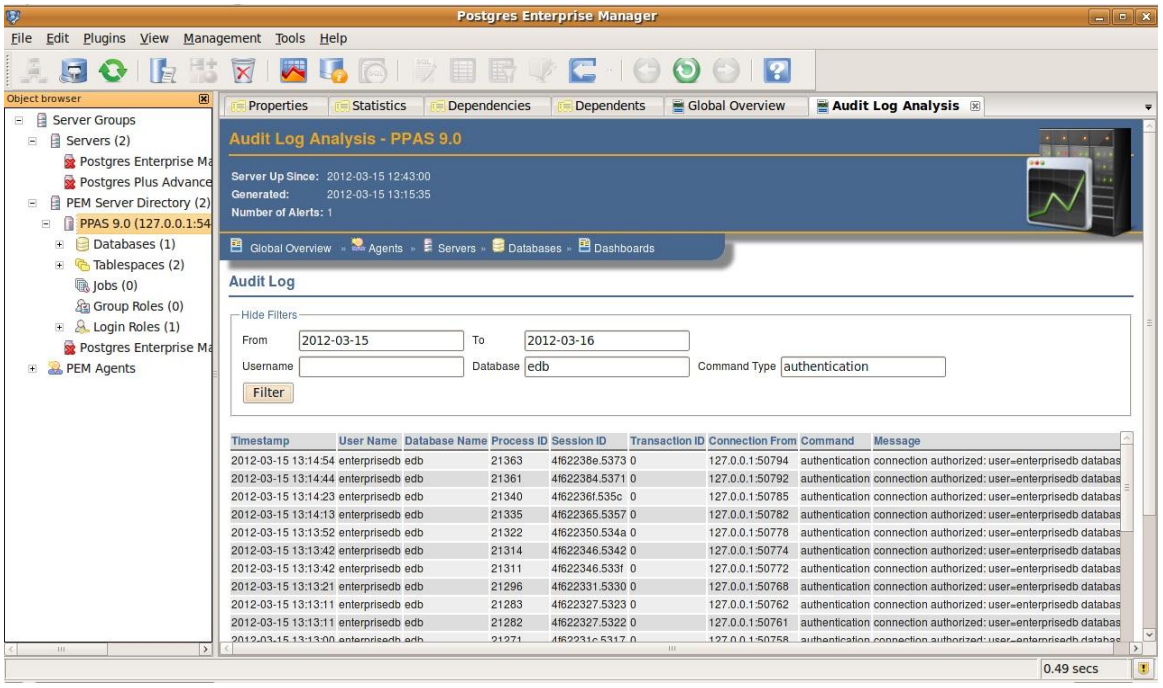

Figure 7.11 – The Audit Log dashboard with filters applied.

The `Audit Log` dashboard then displays records that meet the filtering criteria (see Figure 7.11).

8 SQL Profiling and Analysis

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.

PEM provides the SQL Profiler component to assist in both locating and optimizing poorly-running SQL code. Users of Microsoft SQL Server's Profiler will find PEM's SQL Profiler very similar in operation and capabilities.

8.1 Setup and Configuration

Before using SQL Profiler, you must:

1. Download and install the SQL Profiler product into the managed database instance you wish to profile.

2. Edit the `postgresql.conf` parameter file and 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
   ```

3. Create the functions used by SQL Profiler in your database. The SQL Profiler installation program places a SQL script (called `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\contrib` subdirectory.) You must invoke this script in each database that you will use the SQL Profiler against.

4. Stop and re-start the server for the changes to take effect.

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

8.2 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.

Capturing a new SQL trace is very simple. To open SQL Profiler, select the `SQL Profiler` menu option from the `Management` menu. Upon startup, SQL Profiler will present you with a dialog asking if you would like to start a new trace, or open an existing trace.

If you choose to start a new trace (capturing SQL statements executed against the currently selected server) SQL Profiler will open the `Create New SQL Trace` dialog, as shown in Figure 8.1.

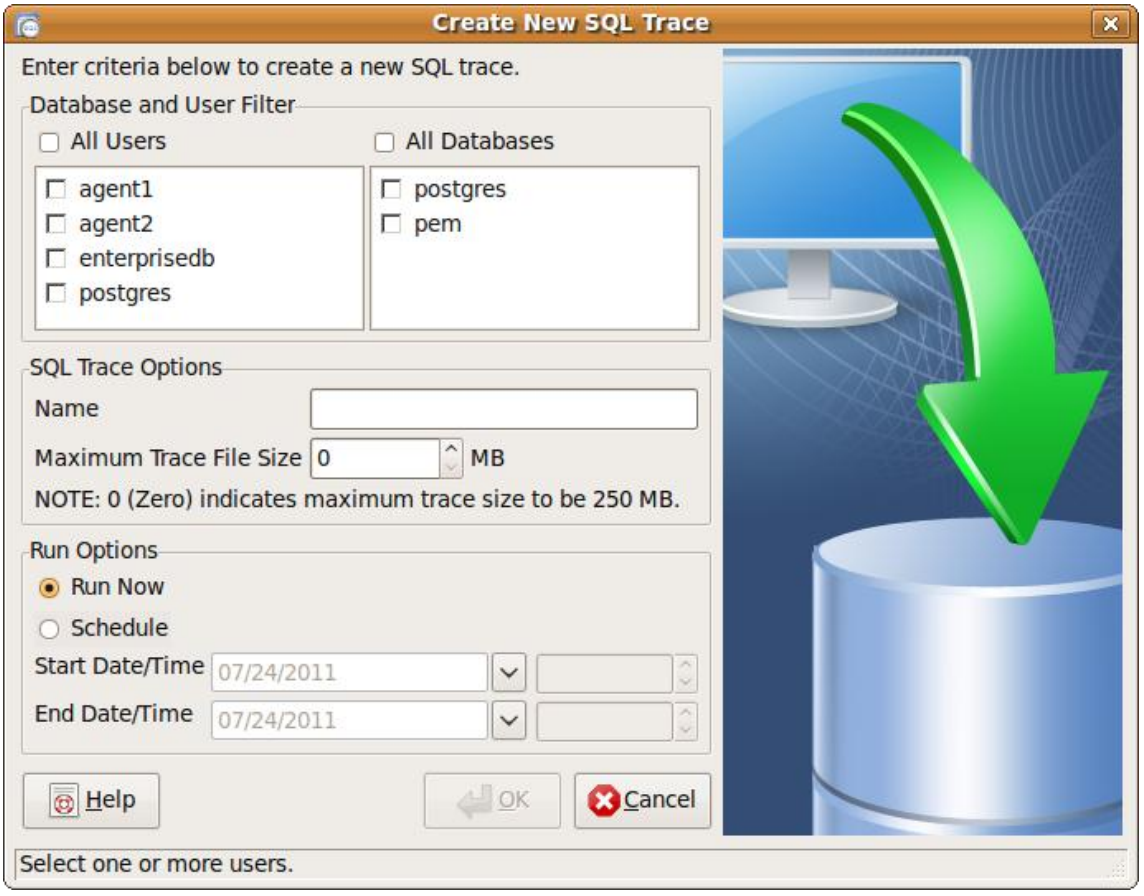

Figure 8.1 - The Create New SQL Trace dialog

Use fields on the dialog to selectively filter the captured SQL statements by user account and by database. You may choose to limit your trace, or to capture every SQL statement sent by all users against all databases.

You can also name your trace for future reference and set a maximum file size for the resulting trace file. Lastly, you can choose to begin capturing SQL immediately in an ad-hoc fashion (if you choose this option, you must manually stop the trace), or you can schedule the trace to run at a later time.

Scheduling a trace is particularly beneficial for capturing workloads during off hours. For example, you may want to capture the SQL statements that are executed against a server from 2 - 4 am. You can configure your trace to run during that timeframe and then examine the output at your leisure when you arrive at work in the morning.

When you are satisfied with all your selections, click the `OK` button to start the trace.

8.3 Analyzing a SQL Trace Output

When you start a new trace, SQL Profiler displays the collected trace data in the top pane of the SQL Profiler dialog. Initially the output will be blank; to see SQL statements that have been captured thus far, click the `Refresh` toolbar icon, or select `Refresh` from the `Trace` menu. If the toolbar is not visible, select `Tool Bar` from the `View` menu to add the toolbar to the display.

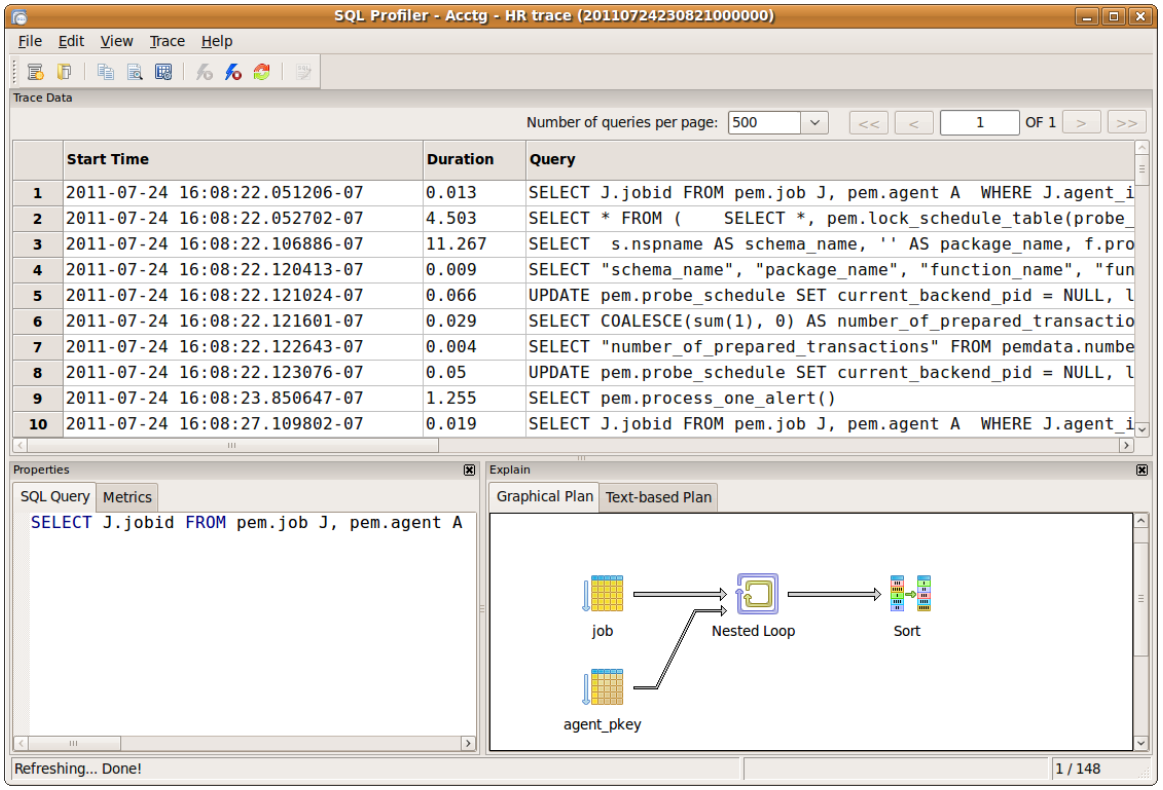

Figure 8.2 - The SQL Profiler trace dialog

The SQL Profiler trace viewer includes three panes, as shown in Figure 8.2:

- The top pane displays SQL statements collected in the trace.
- The lower-left pane displays the full SQL text and statistics for the statement highlighted in the top pane.
- The lower-right pane displays either a graphical or text-based `EXPLAIN` plan for the statement highlighted in the top pane.

SQL Profiler provides a number of features that will help you find poorly-running SQL code on your servers. Some of the most useful options are:

- To display key statistical information collected by SQL Profiler, open the `Select Columns` dialog by selecting the `Columns` option from the `View` menu. The `Select Columns` dialog allows you to customize the columns SQL Profiler displays in the trace output.
- Sorting data in the `Trace Data` pane is easy; just double-click the column header and SQL Profiler will sort the data within the table by that column. Double-click the column header again to reverse the sort order. This allows you to find, for example, the longest running SQL statement very quickly.
- You can filter the contents of a trace to further restrict which SQL statements are displayed. To open the `Trace Filter` dialog, click the `Filter` toolbar button, or select the `Filter` menu option from the `Edit` menu. You can use the `Trace Filter` dialog to construct a filter to display only the SQL statements that match your specific criteria.
- The `Properties` pane displays the complete SQL statement for the currently highlighted row in the `Trace Data` pane; you can use cut and paste functionality to copy the SQL statement to the `Query Tool` SQL IDE for re-working.
- Select the `Metrics` button in the `Properties` pane to display a variety of statistics for the currently highlighted statement. The metrics include information about the number of times a statement has been executed, the overall percentage of execution time vs. all other collected statements, and more.
- The graphical `Explain` pane, provides a graphical interpretation of the execution plan for the selected query. Click a segment of the graph to display statistics for that portion of the query plan.
- To export the contents of a trace to file, choose the `Export` option from the `File` menu. The `Export data to file` dialog allows you to re-format the trace data as it is saved to file.

To stop an active trace, click the `Stop Trace` toolbar button, or select `Stop Trace` from the `Trace` menu.

To view the contents of a previous trace, click on the `Open Trace` toolbar button, or select `Open Trace` from the `File` menu, and select a saved trace. SQL Profiler will close the current trace, and display the newly selected trace.

8.4 Using the Index Advisor

Index Advisor is distributed with Postgres Plus 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.

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

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).

Note again, that the Index Advisor is only included in EnterpriseDB's Postgres Plus Advanced Server, versions 9.0 and above.

9 Postgres Expert - Best Practice Enforcement

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.

Each `Expert` reviews numerous best practice rules written by the database specialists at EnterpriseDB. 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.

9.1 Using the Postgres Expert Wizard

To open Postgres Expert, select the `Postgres Expert` option from the `Management` menu. After welcoming you, the wizard displays the `Experts/Rules` selection dialog (shown in Figure 9.1).

Figure 9.1 - The Experts/Rules selection dialog.

Expand the tree control to review and select the individual rules that will be reviewed by each `Expert`; click `Next` to continue.

The `Servers/Databases` tree control allows you to specify which servers or databases Postgres Expert will review. When you have selected the target servers and databases, click `Next` to continue.

The final Postgres Expert dialog asks you to specify a report destination. You can opt to:

- View the report immediately
- Save the report to file

When you've specified a report destination, click `Finish` to generate a report containing the Postgres Expert recommendations.

9.2 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 (see Figure 9.2).

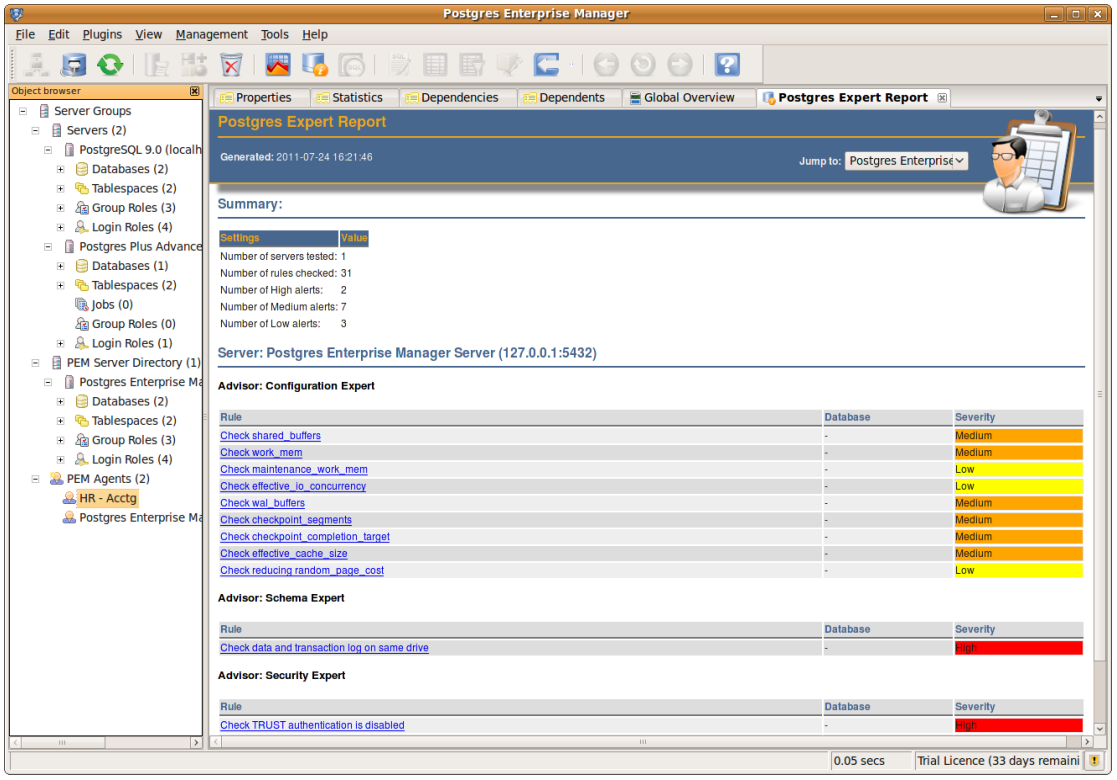

Figure 9.2 - The Postgres Expert report.

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.

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`.

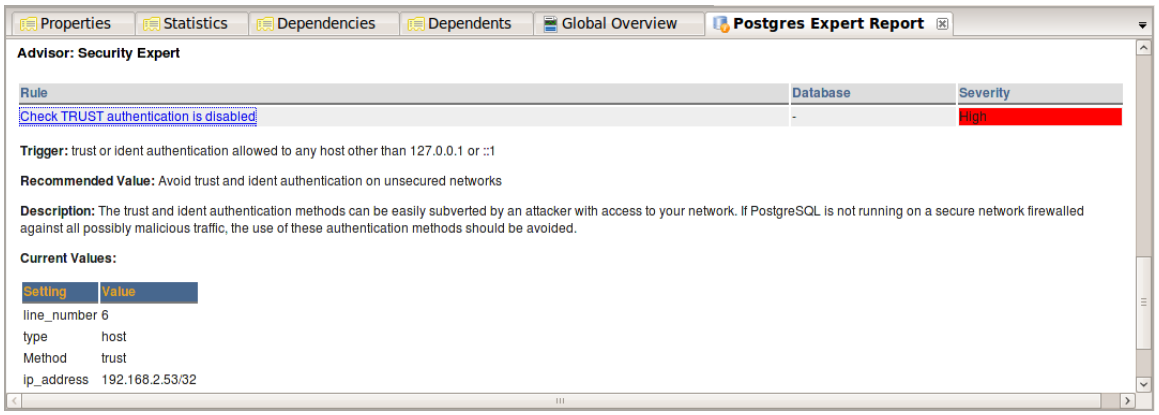

Figure 9.3 - The detailed recommendation for a rule.

Click on each rule to display the best practice recommendation for that rule (see Figure 9.3).

10 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.

10.1 About EnterpriseDB

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. EnterpriseDB's Postgres Plus products are ideally suited for transaction-intensive applications requiring superior performance, massive scalability, and compatibility with proprietary database products. Postgres Plus products provide an economical open source alternative (or complement) to proprietary databases without sacrificing features or quality.

EnterpriseDB understands that adopting a new database is not a trivial task. You have questions that need answers, schedules and budgets to keep, and processes to follow. We have helped thousands of organizations like yours through the steps to investigate, evaluate, prove, develop, and deploy their PostgreSQL solutions.

To make your work easier and faster, we have special self-service sections on our website dedicated to assisting you in each of the steps. Additionally, visit

http://www.enterprisedb.com/why-enterprisedb/postgres-plus-evaluation-production

Getting Started – access to free downloads, installation guides, demos, starter tutorials, and more to help get familiar with the database.

Evaluations and Pilots – learn how EnterpriseDB has helped hundreds of Oracle users cut costs and MySQL users improve operations.

Development – EnterpriseDB employs more PostgreSQL experts, developers and community members than any other company, and offers key application development resources and services.

Deployment – information on how to scale a PostgreSQL application, add Qualities of Service (QoS) like high availability or security, or get a health check.

If you would like to discuss training, consulting, or enterprise support options, please contact EnterpriseDB directly. 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)