

**ORACLE®**

---

**PRIMAVERA**

**Manually Upgrading P6 EPPM for Microsoft SQL Server  
Release 15.1**

May 2015



# Contents

---

<b>Overview of Manually Upgrading a P6 Database to P6 EPPM R15.1 .....</b>	<b>5</b>
<b>Database Upgrade Considerations for P6 EPPM .....</b>	<b>6</b>
Requirements to Manually Upgrade from R8.0 or Later to R15.1 .....	6
Requirements to Manually Upgrade from Version 7.0 to 15.1 .....	6
Tips to Consider When Manually Upgrading P6 EPPM .....	7
Data that is Not Migrated during the P6 EPPM Database Upgrade .....	8
Risks Migration.....	8
<b>Pre-Upgrade Procedures .....</b>	<b>10</b>
Running the Validate Tool for a Microsoft SQL Server Database .....	10
Converting Methodologies to Projects .....	10
<b>Upgrading the P6 EPPM Microsoft SQL Server Database.....</b>	<b>13</b>
<b>Post-Upgrade Procedures .....</b>	<b>16</b>
Creating a Project Template From Projects .....	16
Requirements After the Upgrade from P6 EPPM R8 or later to P6 EPPM R15.1 .....	16
Requirements After the Upgrade from 7.0 to P6 EPPM R15.1 .....	17
<b>Private Database Logins for P6 EPPM .....</b>	<b>19</b>
Resetting Private Database Passwords to Use the New Encryption Algorithm .....	19
Modifying Private Database Logins for P6 EPPM .....	20
<b>Database Administration .....</b>	<b>21</b>
Background Processes and Clean Up in P6 EPPM.....	21
PX Tables Clean Up Procedure .....	22
RDBMS Scheduler Configuration.....	22
Database Settings Table.....	23
Reading Setting Values .....	24
Using Code to Read Setting Values for Oracle .....	24
Using Code to Read Setting Values for Microsoft SQL Server .....	24
Writing Setting Values .....	25
Using Code to Write Setting Values for Oracle .....	25
Using Code to Write Setting Values for Microsoft SQL Server .....	25
Tracking Background Job Execution .....	25
High Level Status Settings .....	25
The BGLOG Table .....	26
SYMON (System Monitor) Procedures.....	27
OBSPROJ_PROCESS_QUEUE Procedure.....	27
USESSION_CLEANUP_EXPIRED Procedure .....	28
Tracking Concurrent Usage of P6 EPPM .....	29
DAMON (Data Monitor) Procedures .....	30

BGPLOG_CLEANUP Procedure .....	30
REFRDEL_CLEANUP Procedure .....	31
REFRDEL Bypass Procedure .....	32
CLEANUP_PRMQUEUE Procedure .....	32
USESSION_CLEAR_LOGICAL_DELETES Procedure .....	33
CLEANUP_LOGICAL_DELETES Procedure .....	34
PRMAUDIT_CLEANUP Procedure .....	35
CLEANUP_USESSAUD Procedure .....	35
USER_DEFINED_BACKGROUND Procedure .....	37
Oracle Database Performance .....	37
Safe Deletes .....	37
Turning Off Safe Deletes .....	38
PRMAUDIT Auditing .....	38
Auditing Level Configuration .....	38
Simple Configuration .....	39
Detailed Configuration .....	39
Auditing Status .....	39
Options Setting .....	40
SETTINGS_WRITE_STRING Procedure .....	41
The Audit Table .....	41
Session Auditing .....	42
Column Audit Data .....	43
Using the Data Pump Utility .....	43
Running a Schema Validation on the Database .....	44
Running the Data Pump Utility Using the Wizard .....	45
Running the Data Pump Utility From the Command Line .....	45
<b>Installing P6 EPPM Applications .....</b>	<b>47</b>
<b>For More Information .....</b>	<b>49</b>
Where to Get Documentation .....	49
Where to Get Training .....	51
Where to Get Support .....	52
<b>Legal Notices .....</b>	<b>54</b>

# Overview of Manually Upgrading a P6 Database to P6 EPPM R15.1

---

To manually convert your P6 version 7.0 or later database to P6 EPPM R15.1, follow the procedures described in this document. Only an experienced database administrator should perform these procedures. Complete the following sections in the order specified:

- ▶ Back up the existing P6 database (also known as PMDB).
- ▶ If you are upgrading from Version 7.0, migrate all your methodologies. You cannot use Project Architect after you upgrade, so you will want to migrate your methodologies before you upgrade. (See **Converting Methodologies to Projects** (on page 10) and **Creating a Project Template From Projects** (on page 16) for manual migration instructions.)
- ▶ Run the appropriate scripts to upgrade the P6 database structure to P6 EPPM R15.1.
- ▶ Perform required and recommended processes after the upgrade is complete. To configure the SQL database server for SSL, use Microsoft's documentation for instructions on configuring SQL Database Server SSL. Direct questions about Installation / Configurations to the database vendor, Microsoft.

---

**Note:** Prior to 15.1, any user who was granted P6 Professional module access could access both P6 Professional and P6 Visualizer applications. When you upgrade your instance of P6 EPPM to the current release, users who have P6 Professional module access will also be granted a separate P6 Visualizer module access. For more information about module access right, see *P6 Reference Manual*.

---

## Database Upgrade Considerations for P6 EPPM

---

You can upgrade your existing P6 EPPM database (version 7.0 and later) to the current release. You must upgrade your P6 EPPM database so it will work with the new version.

### In This Section

---

Requirements to Manually Upgrade from R8.0 or Later to R15.1 .....	6
Requirements to Manually Upgrade from Version 7.0 to 15.1 .....	6
Tips to Consider When Manually Upgrading P6 EPPM .....	7
Data that is Not Migrated during the P6 EPPM Database Upgrade .....	8
Risks Migration.....	8

### Requirements to Manually Upgrade from R8.0 or Later to R15.1

The following list summarizes requirements to upgrade from P6 EPPM R8.0 or later to R15.1:

- ▶ Back up your P6 EPPM database before beginning the upgrade process to ensure you will not lose any data due to unexpected problems. You should back up your database before you perform any of the steps in this document since the upgrade is not reversible. If you are unsure how to back up your database, do not proceed with the upgrade. Contact your database administrator or database vendor for assistance with backing up your database before performing the database upgrade.
- ▶ Manually run all scripts required to upgrade your existing P6 EPPM database.
- ▶ Test the new database to ensure that the upgrade succeeded.

### Requirements to Manually Upgrade from Version 7.0 to 15.1

You must upgrade your P6 EPPM database so it will work with the new version. The following list summarizes the procedures required to manually upgrade your database from P6 EPPM version 7.0:

---

**Notes:** All risk data fields are migrated when upgrading; existing fields are mapped to new fields and others will be mapped to text fields. See **Risks Migration** (on page 8) for more information.

---

- ▶ Back up your P6 EPPM database before beginning the upgrade process to ensure you will not lose any data due to unexpected problems. You should back up your database before you perform any of the steps in this document since the upgrade is not reversible. If you are unsure how to back up your database, do not proceed with the upgrade. Contact your database administrator or database vendor for assistance with backing up your database before performing the database upgrade.

- ▶ See **Data that is Not Migrated during the P6 EPPM Database Upgrade** (on page 8) for details on what information will not migrate.
- ▶ Convert your Methodology Management data to projects before you upgrade to P6 EPPM R15.1. See **Converting Methodologies to Projects** (on page 10) and **Creating a Project Template From Projects** (on page 16) for manual migration instructions..
- ▶ Manually run all scripts required to upgrade your existing P6 EPPM database.
- ▶ Test the new database to ensure the upgrade succeeded.

---

#### Cautions:

- If you are a current Apache JackRabbit user and want to upgrade to P6 EPPM 15.1, JackRabbit documents data will not migrate automatically. Refer to My Oracle Support's Knowledge Articles for information on manually migrating JackRabbit documents to Oracle Universal Content Management. Oracle recommends that you migrate the data before upgrading to 15.1.
  - If you are a current jBPM user and want to upgrade to P6 EPPM R15.1, workflows and reviews data will not be available. You should close all workflows and reviews that are in progress before upgrading to P6 EPPM 15.1. You cannot migrate any of this data to 15.1.
- 

## Tips to Consider When Manually Upgrading P6 EPPM

Please consider the following tips before you manually upgrade P6 EPPM:

### User Name and Password Tips

- ▶ All new users must create a password with at least one character, regardless of whether Password Policy is enabled. P6 EPPM will prompt existing users to follow the new requirement if they reset their password and the Password Policy is disabled.
- ▶ The ADMUSER role must have DBA privileges when upgrading or applying patches.

### Security Tips

- ▶ P6 EPPM includes an encryption algorithm that provides enhanced security for private database logins; however, when you manually configure your database, the database does not automatically enforce the encryption algorithm. If you manually configure your database,  Oracle recommends that you use this encryption algorithm. To do so, you must reset the private database login. See **Resetting Private Database Passwords to Use the New Encryption Algorithm** (on page 19) for instructions. If automatically installing or upgrading your database, you do not need to configure the encryption algorithm. User logins and passwords are not affected.
- ▶ Starting with P6 EPPM R8, P6 performs all user security and administration. Due to this change, all Admin Superusers receive Projects module access during the upgrade. To configure the SQL database server for SSL, use Microsoft's documentation for instructions on configuring SQL Database Server SSL. Direct questions about Installation / Configurations to the database vendor, Microsoft.

### Application Tips

- ▶ P6 EPPM does not support Summary-Only projects. During the P6 EPPM database upgrade, existing Summary-Only projects convert to standard projects, but lose all summary data. You can import the summary project from Microsoft Project into the converted blank project, then summarize the data. See the *P6 Professional Help*.
- ▶ P6 Professional no longer performs recurring job services. Use the Windows command line interface to run batch reports and export projects as a service. To view P6 Professional reports in P6, select the "Store report in Work Products and Documents" option when creating report batches or printing reports. See the *P6 Professional Help* for more information.

### ANSI\_PADDING for a Microsoft SQL Server Database Tips

- ▶ To prevent duplicate values in P6 EPPM modules appearing due to leading or trailing spaces, ensure that the ANSI\_PADDING setting in Microsoft SQL Server is **Off**. For example, if one user enters "Equipment" and another user enters " Equipment" (with a leading space) for resource codes, the database will store both of these as valid values if ANSI\_PADDING is **On**. If ANSI\_PADDING is **Off**, the database will remove the leading space in the second entry and tell the user to enter a unique ID. See the documentation included with Microsoft SQL Server for more information.

## Data that is Not Migrated during the P6 EPPM Database Upgrade

When upgrading P6 EPPM, the following data will not migrate from previous releases:

- ▶ Workflows Repository (supported configuration changed from jBPM to Oracle BPM).
- ▶ Content Repository (if previously an Apache JackRabbit user). Refer to My Oracle Support's Knowledge Articles for information on manually migrating JackRabbit documents to Oracle Universal Content Management.
- ▶ Top Down Estimation, including estimated weights (removed).
- ▶ Invited Users in the Project Workspace (removed).
- ▶ Issue Form Categories (removed).
- ▶ Summary Data (new summary tables incompatible with old summary tables). Refer to My Oracle Support's Knowledge Articles for information on manually migrating Scenarios data to the new summary tables.
- ▶ Methodology Management (replaced with Project Templates). See ***Converting Methodologies to Projects*** (on page 10) and ***Creating a Project Template From Projects*** (on page 16) for manual migration instructions.

## Risks Migration

The following table illustrates the risks data field mapping when upgrading from P6 EPPM database (7.0 or 7.0 service packs) to P6 R15.1.

---

**Note:** Some fields will migrate to text fields instead of new fields because certain fields no longer correspond. The new text fields are noted below.

---

**Risks Fields Migration Table**

<b>P6 EPPM database (version 7.0 and 7.0 service packs) Risks Fields</b>	<b>P6 EPPM R15.1 Risks Field</b>
Risk Id	id
risk_name	name
risk_descr	description
status_code	status
risk_type_id	category_id
risk_control	notes
table_name	table_name
wbs_id	cause (with heading 'Applies to WBS' <WBS name>)
rsrc_id	cause (with heading 'Applies to Resource' <resource name>)
obs_id	cause (with heading 'Responsible Manager' <OBS name>)
priority_type	cause (with heading 'Priority' <priority type>)
add_date	cause (with heading 'Date Identified' <add_date in mmm-dd-yyyy format >)
impact_date	cause (with heading 'Date Identified' <add_date in mmm-dd-yyyy format >)
prbly_pct	cause (with heading 'Probability' <prbly_pct>)
impact_work_qty	cause (with heading 'Impact - Labor Units' <impact_work_qty> - 2 decimals)
impact_equip_qty	cause (with heading 'Impact - Nonlabor Units' <impact_equip_qty> - 2 decimals)
impact_mat_qty	cause (with heading 'Impact - Material Units' <impact_mat_qty> - 2 decimals)
Impact_expense_cost	cause (with heading 'Impact - Expenses' <impact_expense_cost> - 2 decimals)

## Pre-Upgrade Procedures

---

Perform the tasks in the following topics before upgrading your database.

### Running the Validate Tool for a Microsoft SQL Server Database

Use the validate tool to ensure your database is not missing objects that could cause the database to fail to upgrade. You must perform the steps below before you download the 15.1 media pack. If you run the 15.1 validate tool against your current database, it will not work.

- 1) Double-click **validate.bat** (validate.sh for Linux) in your current Database folder of the P6 EPPM physical media or download to start the validate tool.

---

**Note:** Ensure you use the validate file from your current Database folder, not the 15.1 database folder.

---

- 2) Provide the following:
  - a. In the **Database host address** field, enter the server machine name or IP address where you installed Microsoft SQL Server.
  - b. In the **Database host port** field, enter the port number that Microsoft SQL Server is using. The default is 1433.
  - c. In the **Database name field**, enter the name of the existing database that you want to validate.
- 3) View the HTML page that the tool creates to ensure your database is not missing objects.
- 4) If your database is missing objects, do one of the following:
  - ▶ If upgrading from 7.0, go to the 'Client Application' media pack, then locate the \install\database\scripts\ folder to find the scripts you need to fix.
  - ▶ If upgrading from the 8.0, 8.1, 8.2, 8.3, or 8.4 go to the database folder of the P6 EPPM physical media or download, then locate the \database\scripts\install\ to find the scripts you need to fix.

### Converting Methodologies to Projects

To migrate Methodology Management version 7.0 data to P6 R15.1 Project Templates, you must first use Project Architect in the Project Management module to convert the data from a methodology to a project.

To convert Methodology Management data to a project:

- 1) Create a new EPS node in P6 Professional version 7.0 where you can store all your Methodology Management projects.
- 2) Use Project Architect in P6 Professional version 7.0 to create projects from Methodology Management data. For more information on using Project Architect, see version 7.0 of the *Oracle Primavera P6 Project Management Reference Manual*.

---

**Note:** You can create only one project at a time. If you want all of your Methodology Management data moved to P6 Project Templates, contact Oracle Consulting to automate the process.

---

- 3) After you have converted all your Methodology Management data to projects, upgrade P6 EPPM to 15.1.
- 4) Launch P6.
- 5) Convert your upgraded projects (formerly methodologies) to project templates. See ***Creating a Project Template From Projects*** (on page 16).



# Upgrading the P6 EPPM Microsoft SQL Server Database

---

To upgrade the P6 EPPM Microsoft SQL Server Database:

- 1) If you haven't done so, download the **Primavera P6 Enterprise Project Portfolio Management R<release\_level> for <platform>** folder and extract the contents.
- 2) Go to the location of the extracted contents and locate the **P6\_R<release\_level>\p6suite\database** folder.

---

## Notes:

- Make a backup copy of any files you will modify before you edit them.
- You need to execute all the scripts included in this section against PMDB to upgrade.

- 3) Open Microsoft SQL Server Management Studio and log in as the SA user.
- 4) Go to `\scripts\install\PM_<release_level>` and execute the **sspm\_create\_plus\_users.sql** script.

---

## Notes:

- Running the `sspm_create_plus_users.sql` script creates the following user name: PXRptUser. If you have created another user name and wish to use that when running P6 EPPM database scripts, make sure to replace the PXRptUser name with your custom user name in all applicable scripts before running them.
- Do not use special characters in the database name, privileged user, public user, or PX reporting user name, for example: { } [ ] : ; < > , . ? ! @ # \$ % ^ & \* ( ) - \_ | / \ ~ `

- 5) Go to `scripts\common` and execute the **ss\_disable\_triggers.sql** script.
- 6) Go to `\scripts\upgrade\PM_<release_level>` on your local drive. Execute one of the following scripts:

**SSPM\_p70sp1.sql**, if upgrading from P6 version 7.0, 7.0 service pack 1, or 7.0 service pack 2

**SSPM\_p70sp3.sql**, if upgrading from P6 version 7.0 service pack 3

**SSPM\_p70sp5.sql**, if upgrading from P6 version 7.0 service pack 5

**SSPM\_p80.sql**, if upgrading from P6 version 8.0

**SSPM\_p81.sql**, if upgrading from P6 version 8.1

**SSPM\_p81sp3.sql** if upgrading from P6 version 8.1 service pack 3

**SSPM\_p82.sql**, if upgrading from P6 version 8.2

**SSPM\_p82sp1.sql** if upgrading from P6 version 8.2 service pack 1

- SSPM\_p82sp2.sql** if upgrading from P6 version 8.2 service pack 2
- SSPM\_p82sp2f10.sql** if upgrading from P6 version 8.2 service pack 2 Hotfix 10
- SSPM\_p83.sql** if upgrading from P6 version 8.3
- SSPM\_p832.sql** if upgrading from P6 version 8.3 service pack 2
- SSPM\_p833.sql** if upgrading from P6 version 8.3 service pack 3
- SSPM\_p834.sql** if upgrading from P6 version 8.3 service pack 4
- SSPM\_p835.sql** if upgrading from P6 versions 8.3 service pack 5
- SSPM\_p84.sql** if upgrading from P6 version 8.4
- SSPM\_p841.sql** if upgrading from P6 versions 8.4 patch set 1
- SSPM\_p843.sql** if upgrading from P6 versions 8.4 patch set 3

- 7) If you're using version **SSPM\_p70sp1.sql** or later, skip to the next step.

If you're using a version before **SSPM\_p70sp1.sql**, go to \scripts\common and execute the **ss\_update\_usereng.sql** script to grant all users with P6 Professional module access rights to the Projects section in P6.

---

**Note:** Prior to P6 version 7.0, users with Project Management module access also had rights to the Projects section in P6 Web Access. For security reasons, beginning with P6 version 7.0, P6 Professional module access only grants rights to log into the P6 Professional module.

---

- 8) Go to \scripts\common and execute the **ss\_post\_upgrade** script.
- 9) Go to \scripts\upgrade\PM\_<release\_level> and execute the **sspm\_grants.sql** script.

---

**Note:** If you changed the default user names, you must update the privuser, pubuser, and PxRptUser names in this script with your custom user names.

---

- 10) Go to \scripts\upgrade\PM\_<release\_level> and execute the **sspm\_synonyms.sql** script.

---

**Caution:** The **sspm\_synonyms.sql** script contains a table called PUBUSER, do not replace it if you have to change the pubuser user name.

**Note:** If you changed the default user names, you must update the privuser, pubuser, and PxRptUser names in this script with your custom user names.

---

- 11) Go to \scripts\source\PM\_<release\_level> and execute the **sspm\_src.plb** script.

---

**Notes:**

- If you changed the default privuser name, you must update the **sspm\_src.plb** script with your custom privuser name.
  - Do not use special characters in the database name, privileged user, public user, or PX reporting user name, for example: { } [ ] : ; < > , . ? ! @ # \$ % ^ & \* ( ) - \_ | / \ ~ `
-

- 12) Go to \scripts\install\PM\_<release\_level> and execute the **sspm\_database\_version.sql** and **sspm\_querylib.sql** scripts.
- 13) Go to \scripts\common and execute the **ss\_enable\_constraints\_triggers.sql** script.
- 14) Go to \scripts\install\PM\_<release\_level> and execute the **sspm\_init\_bgjobs.sql** script.
- 15) Run the **databaselogins.bat** (on Windows) or **databaselogins.sh** (on Unix) to upgrade the encryption of privilege user passwords.
- 16) Log into the P6 EPPM database as PxRptUser or use your custom PX reporting user name if you created one.
- 17) Go to scripts\source\PM\_<release\_level> and execute the **sspm\_px\_rpt\_src.sql** script.

## Post-Upgrade Procedures

---

Consider the following after you upgrade your database.

### Creating a Project Template From Projects

After you have converted your Methodology Management data to projects and upgraded your database to 15.1, you can create your project templates.

To create a project template:

- 1) Click **Projects**.
- 2) On the **Projects** navigation bar, click **EPS**.
- 3) On the **EPS** page, click the **Actions** ▾ menu and select **Add ▶ Add Project Template**.
- 4) In the **Add Project Template** dialog box, choose the **General** tab.
- 5) On the **General** tab:
  - a. In the **Template ID** field, type in a unique ID for the template.
  - b. In the **Template Name** field, type in a unique name for the template.
  - c. In the **EPS** field, click the corresponding  and choose the EPS where you want to store the template.
  - d. In the **Responsible Manager** field, click the corresponding  and choose a responsible manager.
  - e. In the **Copy from existing project or template field**, click the corresponding  and choose the methodology project you created in **Converting Methodologies to Projects** (on page 10).
  - f. Click **Create**.

### Requirements After the Upgrade from P6 EPPM R8 or later to P6 EPPM R15.1

The following list summarizes the procedures to perform *after* the P6 EPPM upgrade:

#### Backup Procedures

- ▶ Oracle recommends you make a backup of your newly upgraded database.

#### User Name and Password Procedures

- ▶  For security reasons, Oracle strongly recommends that you replace the default database users' (admuser, privuser, pubuser, bgjobuser, and pxrptuser) passwords immediately after a manual database installation or an upgrade from P6 version 7.0. Do not use special characters in the database name, privileged user, public user, or PX reporting user name, for example: { } [ ] ; < > , . ? ! @ # \$ % ^ & \* ( ) - \_ | / \ ~ `

- ▶ For instructions on how to replace the private database login, follow the instructions in ***Modifying Private Database Logins for P6 EPPM*** (on page 20). For all other database user names and passwords, use the tools included with the database. The background job user has to be reset only for manual installations.
- ▶  Oracle recommends using strong passwords. Strong passwords in P6 EPPM contain between 8 and 20 characters and at least one numeric and one alpha character. To further strengthen the password, use a mixture of upper and lower case letters.

### Application Configuration Procedures

- ▶ Install the new P6, P6 Professional, and any additional components.
- ▶ Starting with P6 EPPM R8, all recurring job service functions are hosted by P6. Due to this change, after upgrading to P6 EPPM R8 or later, you must configure Scheduled Services settings in the Primavera P6 Administrator to use this functionality. See the *P6 EPPM Post Installation Administrator's Guide* for details.
- ▶ Assign new or modified privileges to users. See information about security profiles in the *P6 EPPM Post Installation Administrator's Guide*.

### Security Procedures

- ▶  For security reasons, Oracle strongly recommends that you replace the default Admin Superuser (admin) immediately after a manual database installation or an upgrade from P6 version 7.0 and earlier. See the *P6 EPPM Post Installation Administrator's Guide* for details on the default admin superuser.

## Requirements After the Upgrade from 7.0 to P6 EPPM R15.1

The following list summarizes the procedures you need to perform *after* the P6 EPPM upgrade:

### Backup Procedures

- ▶ Oracle recommends that you make a backup of your newly upgraded database.

### User Name and Password Procedures

- ▶  For security reasons, Oracle strongly recommends you replace the default database users' (admuser, privuser, pubuser, bgjobuser, and pxrptuser) passwords immediately after a manual database installation or an upgrade from P6 version 7.0. Do not use special characters in the database name, privileged user, public user, or PX reporting user name, for example: { } [ ] ; < > , . ? ! @ # \$ % ^ & \* ( ) - \_ | / \ ~ `
- ▶ For instructions on how to replace the private database login, follow the instructions in ***Modifying Private Database Logins for P6 EPPM*** (on page 20). For all other database user names and passwords, use the tools included with Oracle Database. The background job user has to be reset only for manual installations.
- ▶  Oracle recommends using strong passwords. Strong passwords in P6 EPPM contain between 8 and 20 characters and at least one numeric and one alpha character. To further strengthen the password, use a mixture of upper and lower case letters.

### Application Configuration Procedures

- ▶ Install the new P6, P6 Professional, and any additional components.

- ▶ Starting with P6 EPPM R8, all recurring job service functions are hosted by P6. Due to this change, after upgrading to P6 EPPM R8 or later, you must configure Scheduled Services settings in the Primavera P6 Administrator to use this functionality. See the *P6 EPPM Post Installation Administrator's Guide* for more information.
- ▶ Resummarize all projects to accurately reflect your summary data. See information about configuring a separate server for job services in the *P6 EPPM Post Installation Administrator's Guide*.
- ▶ Assign new or modified privileges to users. See information about security profiles in the *P6 EPPM Post Installation Administrator's Guide*.
- ▶ All previously defined module access (except Contributor module access (formerly Team Member)), security profiles, OBS access, and resource security settings will still apply to existing users after the database upgrade. For users who had Contributor and P6 Professional module access, they will keep their access during the database upgrade; however, once you edit these users post-upgrade, P6 will remove Contributor module access to comply with P6 EPPM functionality. The upgrade process will remove only Contributor access during the upgrade if users had Contributor and additional module access assignments other than P6 Professional; all other assignments will remain selected. See the *P6 EPPM Post Installation Administrator's Guide* for information on defining module access, security profiles, OBS access, and resource security.
- ▶ During the upgrade to P6 EPPM 15.1, some P6 Activity Views settings will reset. After the upgrade, use the Customize Detail Windows feature to modify the settings that should appear for each view. See the *P6 Help* for information on how to edit Activity Views.
- ▶ Starting with P6 EPPM R8, P6 saves filter definitions globally. Filters still work for Activity Views, but all standard filter assignments reset during the upgrade. Due to this change, views that had Standard Filters applied will show all activities after the upgrade. Reapply filters after the upgrade finishes. See the *P6 Help* for information on how to edit Activity Views.

### Security Procedures

- ▶  For security reasons, Oracle strongly recommends that you replace the default Admin Superuser (admin) immediately after a manual database installation or an upgrade from P6 version 7.0 and earlier. See the *P6 EPPM Post Installation Administrator's Guide* for information on changing the default admin superuser.

## Private Database Logins for P6 EPPM

---

Private database logins are used primarily by administrators to gain direct access to a database. For example, the privileged user login that you use to access the P6 EPPM database is a private database login. You can add, modify, or delete existing logins using the Database Logins tool. This can also be accomplished using the Administration Configuration tool.

P6 EPPM includes an encryption algorithm that provides enhanced security for private database logins; however, when you manually configure your database, the database does not automatically enforce the encryption algorithm. If you manually configure your database,  Oracle recommends that you use this encryption algorithm. To do so, you must reset the private database login. See ***Resetting Private Database Passwords to Use the New Encryption Algorithm*** (on page 19) for instructions. If you automatically configure your database, you do not need to configure the encryption algorithm.

User logins and passwords are not affected.

### Resetting Private Database Passwords to Use the New Encryption Algorithm

Complete the following steps to reset private database passwords to use the new encryption algorithm:

- 1) Go to **P6 EPPM R<release\_level>\database** and run **databaselogins.bat** (databaselogins.sh for Linux).
- 2) On the **Database Connection** dialog box:
  - a. Select the database.
  - b. Type the user name and password of a privileged database user (for example, privuser). This login should have administrative rights on the database.
  - c. Enter the host address, host port, and database/instance name specific to your installation. The Port field displays the default port for the database type you selected.
  - d. Click **Next**.
- 3) On the **Private Database Logins** dialog box:
  - a. Select the private database user name that you wish to reset.
  - b. Highlight the password and change it (or simply re-enter the existing password).
  - c. Click the **Update Password** button.

---

**Note:** To reverse a change, click **Undo**. Undo will reverse any changes made during the current session.

---

- d. Click **Save**.
- e. Click **OK** to exit the Database Logins tool.

## Modifying Private Database Logins for P6 EPPM

Complete the following steps to modify private database logins:

- 1) Go to **P6 EPPM R<release\_level>\database** and run **databaselogins.bat** (databaselogins.sh for Linux).
- 2) On the **Database Connection** dialog box:
  - a. Select the database.
  - b. Type the user name and password of a privileged database user (for example, privuser). This login should have administrative rights on the database.
  - c. Enter the host address, host port, and database/instance name specific to your installation. The Port field displays the default port for the database type you selected.
  - d. Click **Next**.
- 3) On the **Private Database Logins** dialog box:
  - a. Select the private database user name that you wish to modify.
  - b. Enter a new user name.
  - c. Highlight the password, and change it.
  - d. Click the **Update Password** button.

---

**Note:** To reverse a change, click **Undo**. Undo will reverse any changes made during the current session.

---

- e. Click **Save**.
- f. Click **OK** to exit the Database Logins tool.

# Database Administration

Read this chapter to learn how to configure the job scheduler supplied by your RDBMS, how to optimize performance of your P6 EPPM database, and how to configure the PRMAUDIT feature to monitor edits, deletions, and additions to the databases.

## In This Section

Background Processes and Clean Up in P6 EPPM .....	21
PX Tables Clean Up Procedure.....	22
RDBMS Scheduler Configuration .....	22
Database Settings Table .....	23
Reading Setting Values .....	23
Writing Setting Values .....	25
Tracking Background Job Execution .....	25
SYMON (System Monitor) Procedures .....	27
DAMON (Data Monitor) Procedures .....	30
Oracle Database Performance .....	37
Safe Deletes.....	37
PRMAUDIT Auditing.....	38
Using the Data Pump Utility.....	43

## Background Processes and Clean Up in P6 EPPM

Because clean up tasks can be resource intensive and time consuming, in P6 EPPM, these tasks are initiated by two background jobs that run on the database server using the background job processes user name:

- ▶ SYMON (System Monitor), responsible for running procedures that take less than a few seconds to complete.
- ▶ DAMON (Data Monitor), responsible for running procedures that take longer than a few seconds to complete.

Both of these jobs are pre-configured with default settings. Since the default settings are optimal for most environments, you generally do not need to tune them. However, if you need to optimize your background process further, you can use the background job processes user to change the settings to tune the behavior of the background jobs for specific environments.

The background process will:

- ▶ Update the settings table with a record each time a job runs.
- ▶ Update the settings table with a HeartBeatTime record for both the SYMON and DAMON processes. The background processes regularly refresh the record to indicate that they are still running.

When you initiate background jobs:

- ▶ Execute the stored procedure **INITIALIZE\_BACKGROUND\_PROCS**.
- ▶ Locate the logs of background process activity in **BPLOGS** and **SETTINGS** tables within the Primavera database.

### PX Tables Clean Up Procedure

To avoid operational problems for P6, PX table entries are not automatically deleted when you delete data from P6 tables. You can use the following clean up procedure to clear the obsolete entries in PX tables.

- 1) Run the CLEAN\_PX\_DELETE stored procedure to purge obsolete entries in the PX tables:

For a Microsoft SQL Server database:

```
USE [ddname]  
GO  
DECLARE @return_value int  
EXEC @return_value = [dbo].[CLEAN_PX_DELETE]  
SELECT 'Return Value' = @return_value  
GO
```

where *ddname* is the is your admin user name

---

**Note:** You can run this procedure any time you want or you can run it as a database background job and schedule a specific time for it to run.

---

- 2) To delete batch size and max delete from each table in a single run, run the following queries:

---

**Note:** You can specify the delete batch size and max delete from each table in a single run. Delete batch size is the number of rows deleted before a commit. Max delete is the max number of rows that will be deleted from a table in a single run. They have default values if you do not provide your own. They're used to avoid resource problems with the P6 operations.

---

- ▶ `insert into settings (namespace, setting_name, setting_value) values ('PxService.Cleanup', 'DeleteMaxSize', '100000');`
- ▶ `insert into settings (namespace, setting_name, setting_value) values ('PxService.Cleanup', 'DeleteBatchSize', '5000');`

### RDBMS Scheduler Configuration

Since background jobs are initiated by the job scheduler supplied by the RDBMS, you need to ensure that the scheduler for your specific RDBMS is properly configured.

**For a Microsoft SQL Server Database:** P6 EPPM uses the SQL Agent service to automatically schedule background job execution for Microsoft SQL Server. Background jobs handles the maintenance of the utilities tables (for example, USESSION and REFRDEL). As part of background jobs, the agent runs the SYMON and DAMON stored procedures at specific intervals.

## Database Settings Table

### Settings Table Overview

The settings table contains name-value pairs that configure the behavior of the background processes.

### Namespace

The namespace component is a dot-notation string representing a formal path to the parameter.

### Setting Name

The setting name identifies the name of the setting.

### Value

Values in the SETTINGS table are case-sensitive. The value portion of the pair can be:

- ▶ **String.** The string data type is a free text value. The most common string sub-type is interval which represents an interval of time by combining a numeric portion with a unit portion as depicted in the table below.

		Unit portion	Example	Meaning
Numeric portion	+	d	'30d'	Thirty day interval
		h	'2h'	Two hour interval
		m	'10m'	Ten minute interval
		s	'30s'	Thirty second interval

- ▶ **Numeric.** The numeric data type consists of any number.
- ▶ **Boolean.** The Boolean data type can have one of two values: true or false, where zero represents false and any non-zero number represents true.
- ▶ **Date.** The date data type consists of dates.

### Setting Example

The following is an example of a setting:

- ▶ Namespace: database.cleanup.Usession
- ▶ Setting Name: ExpiredSessionTimeout
- ▶ Value: 2h (two hour interval)

### Reading Setting Values

You can configure settings through the Settings API Procedures. These procedures are similar to registry or INI file procedure calls.

#### Reading Settings Values

Use the following SETTINGS\_READ\_\* procedures to determine the current value of specific settings:

- ▶ SETTINGS\_READ\_STRING(ret\_val,namespace,settings\_name,default)
- ▶ SETTINGS\_READ\_DATE(ret\_val,namespace,settings\_name,default)
- ▶ SETTINGS\_READ\_NUMBER(ret\_val,namespace,settings\_name,default)
- ▶ SETTINGS\_READ\_BOOL(ret\_val,namespace,settings\_name,default)

### Using Code to Read Setting Values for Oracle

The following code snippets for the Oracle Database demonstrate how the SETTINGS\_READ\_\* procedures read the setting values.

To retrieve the value of the KeepInterval setting in Oracle:

1) Use the following code:

```
SQL> variable vset varchar2(255)
SQL> exec settings_read_string(:vset,'database.cleanup.Usession',
'ExpiredSessionTimeout');
```

2. The following message should appear:

```
PL/SQL procedure successfully completed.
SQL> print vset
```

### Using Code to Read Setting Values for Microsoft SQL Server

The following code snippets for the Microsoft SQL Server database demonstrate how the SETTINGS\_READ\_\* procedures read the setting values.

To retrieve the value of the KeepInterval setting in SQL:

1) Use the following code:

```
declare @vset varchar(255)
exec settings_read_string @vset
OUTPUT,'database.cleanup.Usession','ExpiredSessionTimeout'
print @vset
```

2) The following message should appear:

```
PL/SQL procedure successfully completed.
SQL> print vset
```

## Writing Setting Values

Use the `SETTINGS_WRITE_STRING` procedure to set the value of a specific setting:

```
SETTINGS_WRITE_STRING(new value, namespace, settings_name);
```

## Using Code to Write Setting Values for Oracle

The following code snippets for Oracle Database demonstrate how the `SETTINGS_WRITE_STRING` procedure sets the value of the `ExpiredSessionTimeout` setting to twelve hours.

To set the value of the `ExpiredSessionTimeout` setting to twelve hours in an Oracle Database:

- 1) Log into SQL \*Plus using `privuser` as your user name.
- 2) Run the following statement:

```
SQL > exec SETTINGS_WRITE_STRING
('12h', 'database.cleanup.Usession', 'ExpiredSessionTimeout');
```

## Using Code to Write Setting Values for Microsoft SQL Server

The following code snippets for Microsoft SQL Server databases demonstrate how the `SETTINGS_WRITE_STRING` procedure sets the value of the `ExpiredSessionTimeout` setting to twelve hours.

To set the value of the `ExpiredSessionTimeout` setting to twelve hours in a Microsoft SQL Server database:

- 1) Open the Query Analyzer/SSMS and connect as `privuser`.
- 2) Select the P6 EPPM database, then run the following statement (using 12 hours as an example):

```
exec SETTINGS_WRITE_STRING '12h', 'database.cleanup.Usession',
'ExpiredSessionTimeout'
```

## Tracking Background Job Execution

You can track the execution of background jobs by monitoring the high level status settings or by inspecting the `BGPLOG` table.

## High Level Status Settings

Each time a job runs it will update the `SETTINGS` table for the `setting_name = 'HeartBeatTime.'` The job can update this value multiple times during the execution. You can monitor the maximum difference between this time and the current date to ensure that the job is running promptly. Refer to the High Level Status Settings table below for information about the `HeartBeatTime` setting.

High Level Status Settings	
<b>Last date and time background job SYMON ran.</b>	
Namespace	database.background.Symon
Setting Name	HeartBeatTime
Default Setting	N/A
<b>Last date and time background job DAMON ran.</b>	
Namespace	database.background.Damon
Setting Name	HeartBeatTime
Default Setting	N/A

### The BGPLOG Table

You can also track the execution of background jobs by inspecting the BGPLOG table. The BGPLOG table holds detailed entries from the background processes including informational, elapsed time, and error entries. Refer to the BGPLOG Table Descriptions for information about what this table contains.

BGPLOG Table Descriptions		
Column	Description	Value
Log_time	Time when background process made a log entry	Datetime
Source	Program generating log entry	"system_monitor", "data_monitor"
Type	Type of message	INFORMATION, ELAPSED TIME, ERROR
Description	Message from the background process	A variable message followed by a number in parenthesis that represents the number of rows that processed. As an example, the message "Complete BGPLOG (40)" indicates that forty rows processed.

## SYMON (System Monitor) Procedures

SYMON runs simple P6 EPPM tasks on a quick schedule. By default, the job runs every minute; the tasks assigned to this job should take only a few seconds to complete on each run. Do not change the scheduled run time (every minute) for this procedure.

### Procedures performed by SYMON

The procedures run by SYMON perform these tasks:

- ▶ Processing the PRMQUEUE entries for Project Security by queuing OBSPROJ updates to the PRMQUEUE table.
- ▶ Marking expired USESSION records as logically deleted.

**Note:** You can manually run queries to assist you with tracking concurrent usage of P6 EPPM.

## OBSPROJ\_PROCESS\_QUEUE Procedure

OBSPROJ\_PROCESS\_QUEUE processes the PRMQUEUE entries for Project Security. It defers processing of OBSPROJ updates by queuing the updates to the PRMQUEUE table.

Refer to the following table for information about the settings associated with the OBSPROJ\_PROCESS\_QUEUE procedure.

### OBSPROJ\_PROCESS\_QUEUE Settings

Setting Description: Maximum project-level queue records to process on each run.	
Namespace	database.obsproj.queue
Setting Name	MaxProjectUpdates
Default Setting	1000
Type	Numeric
Setting Description: Maximum EPS-level queue records to process on each run.	
Namespace	database.obsproj.queue
Setting Name	MaxEpsUpdate
Default Setting	25
Type	Numeric
Setting Description: Maximum times to re-process a failed entry before marking it as an error.	

Namespace	database.obsproj.queue
Setting Name	MaxRetries
Default Setting	50
Type	Numeric

### USESSION\_CLEANUP\_EXPIRED Procedure

USESSION\_CLEANUP\_EXPIRED logically deletes USESSION records that have not updated their last\_active\_time based on the Expired Session settings. Marking expired USESSION records as logically deleted maximizes the number of module access logins that are available. Since it is not cleaning up the underlying data (physically deleting rows), the task completes quickly.

Values in the SETTINGS table control the clean up of expired sessions. By default, although the clean up of expired sessions occurs every two hours, the SETTINGS table does not contain a value for this setting. Use the SETTINGS\_WRITE\_STRING (*value, namespace, setting*) stored procedure to change the default clean up value.

For example, setting the value to "2d" deletes expired sessions older than two days.

**Note:** Oracle recommends that you set the ExpiredLongSessionTimeout sessions to at least one hour longer than your longest job. For example, if your longest job is a summarizer job that usually takes 12 hours, you should set the value in the SETTINGS table to at least 13.

Refer to the table below for information about the USESSION\_CLEANUP\_EXPIRED Settings.

### USESSION\_CLEANUP\_EXPIRED Settings

Setting Description: ExpiredSessionTimeout determines how long an inactive user session will remain in the records before it is marked deleted. User sessions are created when a P6 user logs into P6.	
Namespace	database.cleanup.Usession
Setting Name	ExpiredSessionTimeout
Default Setting	2h
Type	Interval
Setting Description: ExpiredLongSessionTimeout determines how long a session that is running a job-like operation (that is still processing) will remain in the records before it is marked deleted. Job-like operations are processed by job services and some are performed by logged in P6 users. Operations that are considered job-like are:	
<ul style="list-style-type: none"> <li>▶ Scheduling</li> <li>▶ Leveling</li> </ul>	

<ul style="list-style-type: none"> <li>▶ Apply Actuals</li> <li>▶ Update Progress</li> <li>▶ Copy/Paste Project</li> <li>▶ Create Project from Template</li> <li>▶ Maintain Baselines (create new baseline)</li> <li>▶ Approve Timesheets</li> <li>▶ Summarize</li> <li>▶ PX Publish</li> <li>▶ Export</li> <li>▶ Import</li> </ul>	
Namespace	database.cleanup.Usession
Setting Name	ExpiredLongSessionTimeout
Default Setting	12h
Type	Interval

### Tracking Concurrent Usage of P6 EPPM

To track concurrent usage of P6 EPPM, you can run queries against the USESSION and USESSAUD tables to perform self-audits. See sample queries below.

**Note:** See *DAMON (Data Monitor) Procedures* (on page 30) for information on how to set up the USESSAUD procedure. To ensure accuracy of these queries, run them before physically deleting remaining USESSION records and cleaning up the USESSAUD table.

- ▶ Against the USESSION table, run the following query to determine how many users are logged in at a given time:

```
select count(*) from usession where delete_session_id is null
```

- ▶ Against the USESSION table, run the following query to determine how many users are logged into a specific P6 EPPM product at a given time:

```
select count (*) from usession where delete_session_id is null and
app_name=' P6 EPPM product name'
```

where *P6 EPPM product name* is the application abbreviation.

**Note:** You can view all available application abbreviations by running the following query as an administrative database user: `select distinct(db_engine_type) from usereng`

- ▶ Against the USESSAUD table, run a query similar to the following to determine how many users logged into P6 EPPM on a specific date during a specified time range. You can alter the date, time range, and P6 EPPM product as needed. The following example will search for all users who logged into P6 Professional on February 17, 2010 between 9am and 10am:

For a Microsoft SQL Server database:

```
select * from usessaud where login_date between  
'2010-02-17 09:00' and '2011-02-17 10:00' and app_name='Project  
Management'
```

### Tips

See "Counting Users" in the *P6 EPPM Post Installation Administrator's Guide* for information on counting users and how to view the total number of licenses assigned for each module.

## DAMON (Data Monitor) Procedures

The second database job is the DAMON data monitor job. The DAMON job runs the majority of the background processing and is responsible for running background clean up processes required by the application that can potentially take a relatively long time to run.

### Microsoft SQL Server and DAMON

By default, DAMON runs every Saturday. You can set it to run every two weeks or on a specific day. To run DAMON every two weeks, use the following command to set the interval: -eg 2W

To set DAMON to run on a specific day, use the following setting under namespace:  
â€˜Database.background.Damonâ€™ DayOfWeek

### DAMON Procedures

DAMON cleans the:

- ▶ BGPLOG table containing the background logs.
- ▶ REFRDEL table.
- ▶ PRMQUEUE table.
- ▶ Remaining USESSION records.
- ▶ Logically deleted records.
- ▶ PRMAUDIT table.
- ▶ USESSION audit table (USESSAUD).

You can also dynamically extend DAMON functionality via the user-defined procedure, USER\_DEFINED\_BACKGROUND.

## BGPLOG\_CLEANUP Procedure

This procedure keeps the BGPLOG table at a reasonable size. The default clean up interval is 5 days which will result in a table size of about 54,000 records.

Refer to the following table for information about the settings associated with the BGPLOG\_CLEANUP procedure.

**BGPLOG\_CLEANUP Settings**

Setting Description: The oldest records to keep in the BGPLOG table.	
Namespace	database.cleanup.BackGroundProcessLog
Setting Name	KeepInterval
Default Setting	5d
Type	Interval

**REFRDEL\_CLEANUP Procedure**

This procedure deletes records from the REFRDEL table based on the value of the KeepInterval setting. The default setting keeps the REFRDEL records from the last five days.

Refer to the following table for information about the settings associated with the REFRDEL\_CLEANUP procedure:

**REFRDEL\_CLEANUP Settings**

Setting Description: The oldest records to keep in the REFRDEL table.	
Namespace	database.cleanup.Refrdel
Setting Name	KeepInterval
Default Setting	5d
Type	Interval
Setting Description: Identifies the maximum number of minutes before records are deleted from the REFRDEL table.	
Namespace	database.cleanup.Refrdel
Setting Name	DaysToDelete
Default Setting	1
Type	Numeric
Setting Description: Determines the number of minutes that is added to each delete step interval.	
Namespace	database.cleanup.Refrdel
Setting Name	IntervalStep
Default Setting	15
Type	Numeric

## REFRDEL Bypass Procedure

The REFRDEL table maintains a list of deleted records from P6 database tables. However, when an entire project is deleted, a large amount of detailed delete records can be inserted into the REFRDEL table resulting in the potential for downstream performance degradation when joins are made to the REFRDEL table.

The REFRDEL Bypass procedure is an alternative to inserting REFRDEL records for tracking delete records on a large scale. This procedure bypasses the REFRDEL table and simply adds a single delete record for a deleted project and project ID.

By default, the REFRDEL BYPASS procedure is set to 0 (zero).

To enable REFRDEL Bypass, execute the following procedure and set to a non-zero value:

```
SQL> exec SET_REFRDEL_PROJECT_BYPASS(1);
```

Database triggers check for the value of the REFRDEL BYPASS value and process accordingly.

---

**Note:** The bypass procedure is meant only for a PROJECT DELETE operations.

---

## CLEANUP\_PRMQUEUE Procedure

This procedure deletes records from the PRMQUEUE table based on the value of the KeepInterval setting. The remaining settings are similar to the REFRDEL\_CLEANUP.

Refer to the following table for information about the settings associated with the CLEANUP\_PRMQUEUE procedure:

### CLEANUP\_PRMQUEUE Settings

Setting Description: The oldest records to keep in the PRMQUEUE table. Default is five days.	
Namespace	database.cleanup.Prmqueue
Setting Name	KeepInterval
Default Setting	5d
Type	Interval
Setting Description: Determines whether the procedure will delete all of the PRMQUEUE records possible on each pass.	
Namespace	database.cleanup.Prmqueue

Setting Name	DeleteAll
Default Setting	0 (false)
Type	Boolean
Setting Description: Determines whether all of the records are cleaned. If the total record count is less than this number then all the records are cleaned.	
Namespace	database.cleanup.Prmqueue
Setting Name	DeleteAllThreshold
Default Setting	1,000
Type	Numeric
Setting Description: Percentage of records to delete on each pass.	
Namespace	database.cleanup.Prmqueue
Setting Name	DeletePercentage
Default Setting	10(%)
Type	Numeric
Setting Description: Maximum rows to delete on each pass.	
Namespace	database.cleanup.Prmqueue
Setting Name	MaxRowsToDelete
Default Setting	10,000
Type	Numeric

### USESSION\_CLEAR\_LOGICAL\_DELETES Procedure

This procedure physically deletes all logically deleted USESSION records. This procedure does not have settings associated with it: All logically deleted USESSION records are cleared.

## CLEANUP\_LOGICAL\_DELETES Procedure

This procedure removes logically deleted rows based on the value of the KeepInterval setting. Records in the database can be marked as deleted (logically deleted) by setting the DELETE\_SESSION\_ID column to a non-null value. By default, records that were deleted more than 5 days ago will be deleted by this procedure.

### Notes:

- The CLEANUP\_LOGICAL\_DELETES procedure will not delete records whose DELETE\_SESSION\_ID column is set to a negative value.
- This procedure will not delete records older than the earliest user session in USESSION, as determined by the minimum value in the login\_date column.

Refer to the following table for information about the settings associated with the CLEANUP\_LOGICAL\_DELETES procedure:

### CLEANUP\_LOGICAL\_DELETES Settings

Setting Description: The oldest logically deleted records to keep in tables.	
Namespace	database.cleanup.LogicalDelete
Setting Name	KeepInterval
Default Setting	5d
Type	Interval
Setting Description: Determines whether the procedure will delete all of the logically deleted records possible on each pass.	
Namespace	database.cleanup.LogicalDelete
Setting Name	DeleteAll
Default Setting	0 (false)
Type	Boolean
Setting Description: Maximum rows to delete on each pass.	
Namespace	database.cleanup.LogicalDelete
Setting Name	MaxRowsToDelete
Default Setting	10,000
Type	Numeric

## PRMAUDIT\_CLEANUP Procedure

If the auditing feature is enabled, this procedure will physically delete records from the table based on the value of the KeepInterval setting.

Refer to the following table for information about the settings associated with the PRMAUDIT\_CLEANUP procedure:

### PRMAUDIT\_CLEANUP Settings

Setting Description: Should the procedure attempt to clean up PRMAUDIT records.	
Namespace	database.cleanup.auditing
Setting Name	Enabled
Default Setting	1 (true)
Type	Boolean
Setting Description: The oldest audit records to keep in PRMAUDIT.	
Namespace	database.cleanup.auditing
Setting Name	KeepInterval
Default Setting	30d
Type	Interval

## CLEANUP\_USESSAUD Procedure

The CLEANUP\_USESSAUD procedure deletes records from the USESSAUD table based on the KeepInterval parameter setting. All other settings are similar to the REFRDEL\_CLEANUP procedure. The following table describes the settings associated with the CLEANUP\_USESSAUD procedure.

### CLEANUP\_USESSAUD Settings

Setting Description: Determines the oldest records to keep in the USESSAUD table. Based on the logout time, data beyond the KeepInterval parameter value will not be deleted. The KeepInterval parameter setting overrides all other CLEANUP_USESSAUD settings.	
Namespace	database.cleanup.Usessaud
Setting Name	KeepInterval
Default Setting	5d

Type	Interval
Setting Description: Deletes all the REFRDEL records that satisfy the KeepInterval setting on each pass. The DeleteAll parameter setting overrides the settings of DeleteAllThreshold, DeletePercentage, and MaxRowsToDelete parameters.	
Namespace	database.cleanup.Usessaud
Setting Name	DeleteAll
Default Setting	0 (false)
Type	Boolean
Setting Description: Determines the minimum number of records to delete after satisfying the KeepInterval parameter setting. By default, a minimum of 1000 records are deleted. If the total record count is less than this setting, all records are deleted. The DeleteAllThreshold parameter setting overrides the settings of the DeletePercentage and MaxRowsToDelete parameters.	
Namespace	database.cleanup.Usessaud
Setting Name	DeleteAllThreshold
Default Setting	1,000
Type	Numeric
Setting Description: Determines the maximum rows to delete on each pass after satisfying the KeepInterval parameter setting. The MaxRowsToDelete parameter setting overrides the DeletePercentage parameter setting.	
Namespace	database.cleanup.Usessaud
Setting Name	MaxRowsToDelete
Default Setting	10,000
Type	Numeric
Setting Description: Determines the percentage of records to delete on each pass after satisfying the DeleteAllThreshold and MaxRowsToDelete settings. However, the percentage of records deleted is limited to the default value of the MaxRowsToDelete setting.	
Namespace	database.cleanup.Usessaud
Setting Name	DeletePercentage
Default Setting	10 (%)
Type	Numeric

## USER\_DEFINED\_BACKGROUND Procedure

This procedure is an optional customer procedure that DAMON runs. This procedure does not have settings associated with it.

## Oracle Database Performance

You can modify several Oracle database settings to improve the performance of your P6 EPPM database. Run the scripts as described below after you create the database.

### Grant access to the V\_\$TRANSACTION view:

V\_\$TRANSACTION is a system view in Oracle that lists the active transactions in the system.

GET\_SAFETY\_DATE (a procedure in the Project Management schema) accesses this view to get the oldest start time of an active transaction. If the schema owner does not have privileges to this view, then it returns a safety date using the USESSION table. V\_\$Transaction tunes performance during a refresh action. USESSION records the login time of the logged in user, whereas data in the V\_\$Transaction view is recorded at a system level. The logged in user could be logged in for more than an hour (as seen from the USESSION table), but the V\_\$Transaction view has the current transaction datetime, regardless of the time the user logged in.

**Note:** Access to the V\_\$TRANSACTION view was built into the procedure to tune performance with refreshing operations; however, third party functions could impact performance with database refresh operations that use an older time in the V\_\$TRANSACTION view.

To grant access to this view, connect to Oracle as SYS. Run the `RUN_AS_SYS.SQL` script located in the `\Database\scripts\common` folder of the P6 EPPM physical media or download.

### Gather statistics for cost-based optimization:

Oracle 11g and later supports only cost-based optimization, which relies on accurate statistics to determine the optimal access path for a query. To gather the appropriate statistics for the optimizer, which will improve database performance, run the `ORPM_STATS_GATHER.SQL` script located in the `\Database\scripts\common` folder of the P6 EPPM physical media or download.

## Safe Deletes

The P6 EPPM database normally handles restoring select deleted data using a safe delete setting. While using P6 Professional, the Undo command (Edit, Undo) allows users to restore certain types of data that have been deleted. Deleted data remains in the P6 EPPM database until the `CLEANUP_LOGICAL_DELETES` procedure clears it (after 5 days, by default).

See the *P6 Professional Help* for more information about using undo.

### Turning Off Safe Deletes

You can turn off safe deletes to save storage space. Turning off safe deletes disables undo functionality and instantly clears deleted data from the P6 EPPM database.

To turn off safe deletes:

- 1) Verify the current state of your safe deletes setting. In the database, if the table ADMIN\_CONFIG has the following row, a CONFIG\_VALUE of 'N' means turn off safe deletes.  
`CONFIG_NAME = 'SAFEDELETE.ACTIVE' and CONFIG_TYPE = 'SETTINGS'`

**Note:** This is only loaded at startup. If you change CONFIG\_VALUE while a user is running P6 Professional, the setting will not apply until the user restarts the P6 Professional session.

- 2) Once you have determined the current state of your safe deletes setting, run one of the following statements.
  - ▶ To turn off safe deletes for the first time:  
`INSERT INTO ADMIN_CONFIG (CONFIG_NAME, CONFIG_TYPE, CONFIG_VALUE)  
VALUES ('SAFEDELETE.ACTIVE', 'SETTINGS', 'N')`
  - ▶ To turn on safe deletes after it has been turned off:  
`UPDATE ADMIN_CONFIG SET CONFIG_VALUE = 'Y' WHERE CONFIG_NAME =  
'SAFEDELETE.ACTIVE' AND CONFIG_TYPE = 'SETTINGS'`
  - ▶ To turn off safe deletes after it has been turned on:  
`UPDATE ADMIN_CONFIG SET CONFIG_VALUE = 'N' WHERE CONFIG_NAME =  
'SAFEDELETE.ACTIVE' AND CONFIG_TYPE = 'SETTINGS'`
- 3) Restart the P6 server.

### PRMAUDIT Auditing

PRMAUDIT auditing permits you to log the edits, additions, and deletions made by users of P6 EPPM applications. When users make changes, they create a Data Manipulation Language (DML) INSERT, UPDATE, or DELETE statement. PRMAUDIT auditing uses the Data Manipulation Language (DML) INSERT, UPDATE, or DELETE statement being executed against tables in the database schema. Since every application table in the schema has its own auditing trigger, you can log changes made to each table regardless of who made the change or when the change was made. The database schema owner owns the auditing trigger: you can bypass trigger execution.

### Auditing Level Configuration

You can adjust the amount of information that is logged by adjusting the audit level for each table. You can refine the audit further by setting the audit level individually for insert, updates, and deletes within each table.

## Auditing Levels

Level	Description
Level 0	No audit.
Level 1	Row-level audit. Audit only the operation without column details
Level 2	Column-level Audit without blobs. Audit changes to the data at the column level but without blob changes
Level 3	Full Audit. Audit changes to the data at the column level. For Microsoft SQL server, column level changes to blobs are not included.

## Simple Configuration

You can use two configuration procedures to provide simple control of the auditing feature:

- ▶ `auditing_enable(table_name, level)`
- ▶ `auditing_disable(table_name)`

You can set the audit level on an individual table or the same audit level for all of the tables. However, the simple configuration procedures do not allow for setting individual auditing levels for insert, update, or delete operations within a table.

## Detailed Configuration

You can configure auditing trigger behavior by changing values in the settings table that enable or disable:

- ▶ The auditing feature itself
- ▶ The auditing of specific tables
- ▶ The auditing of table insert, update, or delete operations within each table

## Auditing Status

You can enable or disable the auditing feature by using the `database.audit.Enable` setting. Use the `settings_write_bool` procedure to enable/disable the auditing feature.

### Microsoft SQL Server Example:

To enable the auditing feature in Microsoft SQL Server, use the following code:

```
exec settings_write_bool 1, 'database.audit', 'Enabled'
```

## Options Setting

Each table's auditing settings are controlled by the Options setting in each table's auditing namespace (for example, database.audit.TASK). The Options setting is a three character string with a numeric value in each character position representing the audit level for insert, update, and delete.

Auditing Level Options Setting by Table Operation				
	Operation			
	Insert	Update	Delete	Description
Level	0	0	0	No audit.
	1	1	1	Row-level audit. Audit only the operation without column details.
	2	2	2	Column-level audit without blobs. Audit changes to the data at the column level but without blob changes.
	3	3	3	Full Audit. Audit changes to the data at the column level. For Microsoft SQL server, column level changes to blobs are not included.

The following table provides some examples of the options setting:

Setting the Auditing Level Options Setting by Table Operation Examples			
Namespace	Setting	Value	Description
database.audit.TASK	Options	330	Fully audit any insert and update operations. Do not audit any delete operations.
database.audit.PROJWBS		001	Row-level audit on deletes only.
database.audit.TASKRSRC		333	Fully audit.

## SETTINGS\_WRITE\_STRING Procedure

You can change table audit settings using the `settings_write_string` procedure.

### Microsoft SQL Server Example:

To set the table settings to fully audit insert and update operations but ignore any delete operations, use the following code for Microsoft SQL Server:

```
exec settings_write_string '330','database.audit.TASK','Options'
```

**Note:** Changes to auditing settings will not appear immediately in the application. The program will need to close the database connection and then reconnect to the database to get the new settings.

## The Audit Table

Audit records are inserted into the PRMAUDIT table. One record is inserted into the audit table for each row changed in the database.

PRMAUDIT Table		
Column	Type	Description
audit_date	Date	Date and time of change
table_name	String(30)	Table Name
pk1, pk2, pk3, pk4	String(255)	Primary key values for audited record
oper	String(1)	I=Insert, U=Update, D=Delete
prm_user_name	String(32)	P6 EPPM user name if the change was made in P6 EPPM applications
audit_info	String(4000)	Column changes up to 4000 characters (Level 2 and 3 only)
audit_info_extended	BLOB	Blob changes and overflow from audit_info (Level 2 and 3 only)
logical_delete_flag	String(1)	Flag for deletes that are logical (marked) rather than a physical delete
rdbms_user_name*	String(255)	Database user name (usually privuser)
os_user_name*	String(255)	Operating system user name of connected session <b>Note:</b> When auditing is enabled against a SQL Server database,

		PRMAUDIT.os_user_name will always return a null value. This is because the value for os_user_name in PRMAUDIT table is taken from the nt_username column in master.sys.sysprocesses view and the nt_username value in master.sys.sysprocesses view is empty for the P6 Professional program because P6 is connected to SQL DB using SQL authentication. The nt_username field is filled only when we connect to SQL DB using Windows authentication.
program*	String(255)	Name of program connecting to the database
host_name*	String(255)	Computer name of connected session
app_name*	String(25)	Name of application connected to the database
netaddress*	String(24)	IP or MAC address of connected session

---

**Note:** Grant select privileges to the administrative user on V\_\$SESSION to ensure correct values for several auditing table values.

---

## Session Auditing

Activity for the USESSION table is audited with its own trigger and table. When an application user logs out of the system they logically delete, or mark, their session record in the USESSION table. One record is written to the USESSAUD table for each logout. The format of the USESSAUD table mirrors that of the USESSION table. This audit can be enabled using the usessaud\_enable procedure and disabled using the usessaud\_disable procedure.

## Column Audit Data

The data changes for each audit are stored in the `audit_info` and `audit_info_extended` columns. The `audit_info` column contains all the row changes as long as they do not exceed 4000 characters. Changes over 4000 characters or any edit to a blob will be written to the `audit_info_extended` BLOB column.

Data in the two `audit_info` columns has a specific format. Each column audit within the data begins with either ":O" (old data) or ":N" (new data) to distinguish between the audit of the previous (old) or the changed (new) value (for BLOB columns the data starts with :BLOBO or :BLOBN). Directly after this is the name of the column in lowercase. Following the column name is the length of the audited value in a fixed four character field. Finally the actual data is placed in the audit record. Updates will have both an old and new value for each change. Inserts will have only a new value and deletes only an old value.

The following is an example of the audit record for TASK to change the `task_code` from 'A1010' to 'B102':

```
audit_info =>:Otask_code: 5:A1010:Ntask_code: 4:B102
```

## Using the Data Pump Utility

The data pump utility is a java-based tool that enables you to pump data from one database to another at its current version only. Data can be pumped from the following types of databases:

- ▶ SQL Server to Oracle
- ▶ Oracle to SQL Server
- ▶ Oracle to Oracle
- ▶ SQL Server to SQL Server

Depending on the operating system, use the appropriate file to run the data pump utility:

- ▶ For Windows operating systems, use `migrate.bat`.
- ▶ For Linux/Unix operating systems, use `migrate.sh`.

Use the data pump utility to:

- ▶ only import data into a newly created database
- ▶ repair damaged databases to only correct issues with regard to database objects (such as constraints, views)

Do **not** use the data pump utility to:

- ▶ upgrade from one database version to another
- ▶ import data into an existing database
- ▶ correct data-specific issues
- ▶ correct schema related issues. It is not ideal for large databases (10GB+) due to the performance of the utility. For larger databases, try to manually resolve schema related issues first whenever possible.

### Prerequisite

**Running a Schema Validation on the Database** (on page 44).

### Running a Schema Validation on the Database

The schema validation utility finds missing, extra, and modified schema objects for a P6 database. Depending on the operating system, use the appropriate file to run the schema validation.

- ▶ For Windows, run `validate.bat`
- ▶ For Linux/Unix systems, run `validate.sh`

To run the schema validation, proceed as follows.

- 1) In the software download, locate the utility in the **Database Download** folder.
- 2) Edit the utility similar to the following:

```
@echo off
REM -----
REM Run the Primavera Schema Validation Tool
REM -----

SET JAR_FILE_DIR=lib
SET JAR_FILE_NAME=dbmt.jar
SET DB_SCHEMA=ppm

SET JVM_PROPERTIES=-Ddbmt.dbschema.package.location=%JAR_FILE_DIR%
-Dprimavera.db.schema=%DB_SCHEMA%
-Dcom.primavera.database.logfile=SchemaValidation.log
SET DBMT_ACTION=application/ppmschemaval
SET DBMT_COMMAND=schemavalpm
set JAVA_HOME=C:\Program Files\Java\jre6
IF NOT EXIST "%JAVA_HOME%\bin\java.exe" (
    echo JAVA_HOME IS NOT SET
    pause
    goto :EXIT
)
```

- 3) Run `validate.bat` (or `validate.sh`).

For an Oracle database in an Enterprise environment, enter values for the following:

- ▶ **Username:** Use the appropriate ADMUSER username and password.
- ▶ **Database host address:** Database server name or IP address.
- ▶ **Database host port:** The port your Oracle listener is listening on. (Default is 1521)
- ▶ **Database name (SID):** Enter the Oracle SID for your database.

---

**Note:** When running for an Enterprise version of the database, you will be prompted for the appropriate privileged and public usernames.

---

For a SQL Server 2005 database, enter values for the following:

- ▶ **Database host address:** Database server name or IP address.
- ▶ **Database host port:** The SQL port.
- ▶ **Database name:** pmdb\$primavera (default)

---

**Note:** When running for an Enterprise version of the database, you will be prompted for the appropriate privileged and public user names.

---

- 4) The browser displays the results of the schema validation utility.

### Running the Data Pump Utility Using the Wizard

Depending on the operating system, use the appropriate file to run the data pump utility.

- ▶ For Windows, run `migrate.bat`
- ▶ For Linux/Unix systems, run `migrate.sh`

---

**Note:** Do not use `migrate.bat` to convert database type from EPPM to PPM or vice-versa.

---

To run the data pump utility using the wizard, proceed as follows:

- 1) In the `\install\database` directory, double-click the `migrate.bat` (or `migrate.sh`) utility.

You must launch this utility from within the `\install\database` directory, because it has dependencies to other files inside this directory.

- 2) Follow the prompts on each screen by providing the connection information for the source and target database being migrated.

---

**Note:** When using an Oracle schema, the migration utility will allow you to create new tables or use existing tables in the target instance, but new schema users will have to be created through the migration process. The ability to map to existing schema users is not currently available through the migration process.

---

### Running the Data Pump Utility From the Command Line

Depending on the operating system, use the appropriate file to run the data pump utility.

- ▶ For Windows, run `migrate.bat`
- ▶ For Linux/Unix, run `migrate.sh`

---

**Note:** Do not use `migrate.bat` to convert the database type from EPPM to PPM or vice-versa.

---

To run the data pump utility from the command line, proceed as follows.

- 1) Open a new command line console.
- 2) Change directory to where the utility is located.

### 3) Enter appropriate parameters to define the migration.

#### On a Windows Operating System

##### ▶ SQL Server to Oracle Data Migration Syntax

```
Migrate.bat -source sa/sa@sqlserver:rcgsrv:1433:vader_pmdb -target system/manager@oracle:rcgsrv:1521:rcg02 -db PM
```

##### ▶ Oracle to SQL Server Data Migration Syntax

```
Migrate.bat -source admuser/admuser@oracle:rcgsrv:1521:rcg02 -target sa/sa08@sqlserver:rcgsrv:1433:vader_pmdb -db PM
```

##### ▶ SQL Server to Oracle Data Migration Syntax Creating Non-Default Oracle Users

```
Migrate.bat -source sa/sa@sqlserver:rcgsrv:1433:vader_pmdb -target system/manager@oracle:rcgsrv:1521:rcg02 -db PM -admuser admjeff -privuser privjeff -pubuser pubjeff
```

#### On a Unix/Linux Operating System

##### ▶ SQL Server to Oracle Data Migration Syntax

```
sh migrate.sh -source sa/sa@sqlserver:saumverm-lap\primavera:1433:PMDB -dattbsp <Name of data tblspc at target> -ndxtbsp <Name of index tblspc at target> -pxtbsp <Name of px tblspc at target> -lobtbsp <Name of LOB tblspc at target> -admuser <admuser at target> -admpass <admuser password at target> -privuser <privuser at target> -privpass <privuser password at target> -pubuser <pubuser at target> -pubpass <pubuser password at target> -pxrptuser <pxrptuser at target> -pxrptpass <pxrptuser password at target> -bgjobuser <bgjobuser at target> -bgjobpass <bgjobuser password at target> -target system/admin@oracle:oldb-orcl:1521:PMDB
```

For example,

```
sh migrate.sh -source sa/sa@sqlserver:saumverm-lap\primavera:1433:PMDB -dattbsp PMDB_DAT1 -ndxtbsp PMDB_NDX1 -pxtbsp PMDB_PX_DAT1 -lobtbsp PMDB_LOB1 -admuser admuser -admpass admuser -privuser privuser -privpass privuser -pubuser pubuser -pubpass pubuser -pxrptuser pxrptuser -pxrptpass pxrptuser -bgjobuser bgjobuser -bgjobpass bgjobuser -target system/admin@oracle:oldb-orcl:1521:PMDB
```

##### ▶ Oracle to SQL Server Data Migration Syntax

```
sh migrate.sh -source privuser/privuser@oracle:oldb-orcl:1521:PMDB -privuser <privuser at target> -privpass <privuser password at target> -pubuser <pubuser at target> -pubpass <pubuser password at target> -pxrptuser <pxrptuser at target> -pxrptpass <pxrptuser password at target> -target sa/sa@sqlserver:jdoe-lap\primavera:1433:PMDB
```

For example,

```
sh migrate.sh -source privuser/privuser@oracle:oldb-orcl:1521:TEST -privuser privuser -privpass privuser -pubuser pubuser -pubpass pubuser -pxrptuser pxrptuser -pxrptpass pxrptuser -target sa/Primavera1@sqlserver:jdoe-lap\primavera:1433:test
```

---

## Installing P6 EPPM Applications

---

Run **setup.exe** in the Primavera P6 Enterprise Project Portfolio Management physical media or download to install the following P6 EPPM applications:

---

**Note:**

- You do not have to install all applications; you can pick and choose what you install. However, you must install P6 to run the other P6 EPPM applications.
- Oracle recommends that you install P6 and P6 Team Member separately so that you can create an Admin configuration for both of them. If you install them together, you will only be able to create an Admin configuration for P6.

- 
- ▶ P6 Team Member (includes P6 Team Member Web and E-mail Statusing Service)

---

**Notes:**

- You will need to configure E-mail Statusing Service separately. The installation wizard will provide the E-mail Statusing Service files and you will use the config.properties file to set up this service. See the *P6 EPPM Post Installation Administrator's Guide*.
- P6 mobile supports SSL (https) only when it has a certificate signed by a valid authority.

- 
- ▶ P6 EPPM Web Services
  - ▶ P6 Professional Cloud Connect

Select the P6 Professional Cloud Connect install option if you plan to use P6 Professional with an EPPM Oracle database on the cloud by connecting to a remote service. The wizard sets up a remote method of access to connect to the EPPM database and the P6 APIs. For more details on establishing the P6 Professional Cloud Connect database connection in P6 Professional, see the *P6 Professional for EPPM Installation and Configuration Guide*.

---

**Note:** The wizard will also install P6 Integration API (in remote mode) by default.

- 
- ▶ P6 Integration API (remote mode only)

---

**Note:** Oracle provides a separate installer for P6 Integration API local mode. See the *P6 Integration API Configuration Guide*.

- 
- ▶ P6 mobile

---

**Note:**

---

- P6 mobile apps are downloaded through the Apple App Store or the Google Play Store. See the *P6 Team Member User's Guide*.
- P6 mobile supports SSL (https) only when it has a certificate signed by a valid authority.

Based on whether you are installing or upgrading, see one of the following books:

- ▶ *Installing and Configuring P6 EPPM*
- ▶ *Upgrading and Configuring P6 EPPM*

## For More Information

### Where to Get Documentation

Complete documentation libraries for P6 EPPM releases are available on the Oracle Technology Network (OTN) at:

<http://www.oracle.com/technetwork/documentation/primavera-093289.html>

From this location you can either view libraries online or download them to have local copies. We recommend viewing them from OTN to ensure you always access the latest versions, including critical corrections and enhancements.

P6 EPPM is configured to access its help systems on OTN. However, you can also install local versions when you install the software.

The documentation assumes a standard setup of the product, with full access rights to all features and functions.

The following table describes the core documents available for P6 EPPM and lists the recommended readers by role. P6 EPPM roles are described in the *Planning Your P6 EPPM Implementation* guide.

Title	Description
<i>What's New in P6 EPPM</i>	<p>Highlights the new and enhanced features included in this release.</p> <p>You can also use the <i>P6 EPPM Cumulative Feature Overview Tool</i> to identify the features that have been added since a specific release level.</p> <p>All users should read this guide.</p>
<i>Planning Your P6 EPPM Implementation</i>	<p>Explains planning your implementation and provides an installation process overview, frequently asked questions, client and server requirements, and security information.</p> <p>The P6 EPPM network administrator/database administrator and P6 administrator should read this guide.</p>
<i>P6 EPPM Installation and Configuration Guide</i>	<p>Explains how to install and configure P6 EPPM using the P6 EPPM Installation and Configuration wizards.</p> <p>The P6 EPPM network administrator/database administrator and P6 administrator should read this guide.</p>

Title	Description
<i>P6 EPPM Installation and Manual Configuration Guide</i>	<p>Explains how to install and configure the P6 EPPM using the P6 EPPM Installation wizards, and how to manually configure individual components.</p> <p>The P6 EPPM network administrator/database administrator and P6 administrator should read this guide.</p>
<i>P6 EPPM Post Installation Administrator's Guide</i>	<p>Describes how to get started using P6 EPPM applications after you have installed and configured them. Complete the tasks in this guide before letting your users work with these applications. These tasks include information about configuring your users and security settings and privileges, configuring your Primavera P6 Administrator settings, and finalizing your P6 Integration API and P6 EPPM Web Services settings.</p> <p>The P6 EPPM network administrator/database administrator and P6 administrator should read this guide.</p>
<i>Tested Configurations</i>	<p>Lists the configurations that have been tested and verified to work with P6 EPPM.</p> <p>The network administrator/database administrator and P6 EPPM administrator should read this document.</p>
<i>P6 User's Guide</i>	<p>Explains how to plan, set up, and manage projects in a multiuser environment. If you are new to P6, start with this guide to learn how to use the software effectively to plan and manage projects. When you need more detail, refer to the P6 Help.</p> <p>The program manager, project manager, resource/cost manager, team leader, and all P6 users should read this guide.</p>
<i>P6 Help</i>	<p>Explains how to use P6 to administer, plan, set up, and manage projects, portfolios, workflows, timesheets, documents, and reports in a multiuser environment. Describes how to analyze performance and ROI, and analyze budgets. If you are new to P6, use this Help to learn how to use the software effectively.</p> <p>The operations executive, P6 EPPM and P6 administrator, program manager, project manager, resource/cost manager, team leader, and all users should read this Help.</p>
<i>P6 Data Dictionary</i>	<p>Defines fields used in P6.</p> <p>All P6 users should refer to this guide if they need a field definition.</p>

Title	Description
<i>P6 Team Member Web Help</i>	Describes how to use P6 Team Member Web to provide status on activities. P6 Team Member Web users should read this Help.
<i>P6 EPPM Web Services Programmer's Guide</i>	Describes how to invoke, use, and troubleshoot the available services and operations within supported environments. When you need specific information about the services and operations available, refer to the P6 EPPM Web Services Reference Manual. Anyone who wants to develop applications that interact with P6 should read this guide.
<i>P6 EPPM Web Services Reference Manual</i>	Describes all services and operations available in P6 EPPM Web Services. Anyone who wants to develop applications that interact with P6 should read this guide.
<i>P3 to P6 EPPM Migration Guide</i>	Provides best practices for migrating your P3 data to P6 EPPM and details how P3 functionality maps to P6 EPPM functionality. All administrators should read this guide if your organization is moving from P3 to P6.

### Distributing Information to the Team

You can copy the online documentation to a network drive for access by project participants. Team members can then view or print those portions that specifically relate to their roles in the organization.

Throughout this documentation, the Security Guidance icon  helps you to quickly identify security-related content to consider during the installation and configuration process.

### Where to Get Training

To access comprehensive training for all Primavera products, go to:

<http://education.oracle.com>

### Oracle Learning Library

The Oracle Learning Library (OLL) provides online learning content covering Primavera products. Content includes whitepapers, videos, tutorials, articles, demos, step-by-step instructions to accomplish specific tasks, and self-paced interactive learning modules.

To access the learning library's Primavera content, go to:

<http://www.oracle.com/oll/primavera>

### Where to Get Support

If you have a question about using Oracle products that you or your network administrator cannot resolve with information in the documentation or help, visit <http://support.oracle.com/>. This page provides the latest information on contacting Oracle Global Customer Support, knowledge articles, and the support renewals process. For more information about working with Support, visit <https://support.oracle.com/epmos/faces/DocumentDisplay?id=888813.2> to view **Support Tools & Tips**.

The following knowledge articles are a good place to start your research because they link to the most frequently referenced articles about P6 EPPM

- ▶ Primavera Product Master Notes [ID 1489367.1]
- ▶ Master Note For Primavera P6 Common Application Questions Or Issues [ID 1292929.1]

P6 EPPM integrates with different Oracle applications; when you create a Service Request, be sure to open the request with the proper Support team. To ensure you reach the proper Support team, enter the correct product information when you create the Service Request. Each product has its own support line.

- ▶ Use the **Primavera P6 EPPM** support line when you are having installation, configuration, or connection issues related to P6 EPPM.
- ▶ Use one of the following support lines when you are having installation or configuration issues that do not relate to P6 EPPM.
  - ▶ Oracle WebLogic Server
  - ▶ Oracle Database Server
  - ▶ BI Publisher
  - ▶ BPM
  - ▶ Oracle Webcenter Content Core Capabilities (formerly Universal Content Management)
  - ▶ Oracle Enterprise Manager
  - ▶ Oracle Access Manager
  - ▶ Oracle AutoVue

### Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/us/support/contact-068555.html> or visit <http://www.oracle.com/us/corporate/accessibility/support/index.html> if you are hearing impaired.

### Using Primavera's Support Resource Centers

Primavera's Support Resource Center provides links to important support and product information. Primavera's Product Information Centers (PICs) organize documents found on My Oracle Support (MOS), providing quick access to product and version specific information such as important knowledge documents, Release Value Propositions, and Oracle University training. PICs also offer documentation on Lifetime Management, from planning to installs, upgrades, and maintenance.

Visit <https://support.oracle.com/epmos/faces/DocumentDisplay?id=1486951.1> to access links to all of the current PICs.

PICs also provide access to:

- ▶ **Communities** which are moderated by Oracle providing a place for collaboration among industry peers to share best practices.
- ▶ **News** from our development and strategy groups.
- ▶ **Education** via a list of available Primavera product trainings through Oracle University. The Oracle Advisor Webcast program brings interactive expertise straight to the desktop using Oracle Web Conferencing technology. This capability brings you and Oracle experts together to access information about support services, products, technologies, best practices, and more.

## Legal Notices

---

Oracle Primavera Manually Upgrading P6 EPPM for Microsoft SQL Server

Copyright © 1999, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

**U.S. GOVERNMENT END USERS:** Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate failsafe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

This software or hardware and documentation may provide access to or information on content, products and services from third-parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.