Copyright

Copyright © 2005, 2011, Oracle and/or its affiliates. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

The platform-specific hardware and software requirements included in this document were current when this document was published. However, because new platforms and operating system software versions might be certified after this document is published, review the certification matrix on the My Oracle Support (formerly OracleMetaLink) Web site for the most up-to-date list of certified hardware platforms and operating system versions. The My Oracle Support (formerly OracleMetaLink) Web site is available at the following URL:

http://metalink.oracle.com/

or

http://support.oracle.com/

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable: U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software -- Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.
The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle and Primavera are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners. The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

To view the P6 Commercial Notices and Disclosures for Documentation, go to the \Documentation\<language>\Notices and Disclosures folder of the P6 physical media or download.
The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.
Contents

Business Problem ........................................................................................................... 7
  P6 Extended Schema vs. ODS .................................................................................... 7
Overview .......................................................................................................................... 9
  P6 Extended Schema .................................................................................................. 11
  Publication Services .................................................................................................... 11
Configuration, Sizing and Performance ................................................................. 17
  Configuring and Deploying Publication Services .................................................... 17
  Recommended Hardware Configuration ................................................................. 29
  Performance and Scalability Considerations ......................................................... 30
Reporting ......................................................................................................................... 37
  Report Security ........................................................................................................... 37
  Upgrade ..................................................................................................................... 38
  Creating Reports ........................................................................................................ 39
Eventing ........................................................................................................................... 43
P6 Extended Schema FAQ ............................................................................................ 45
Business Problem

Companies face an ever increasing need for access to the most up to date data regarding their business. Over the years, P6 has been addressing this need with enhanced data access functionality, such as the introduction of the Enterprise Reporting Database in P6 Release 6.0 and integration with the Oracle Business Intelligence suite in Release 8.0. In P6 Release 8.1, we have gone one step further by extending the P6 database schema to include the storage of logical or calculated fields. This approach has a number of key advantages:

- It allows for near real-time reporting from the core P6 database schema.
- It gives customers more control over their data.
  - The administrator can choose which project and global data to calculate and store, and how often.
- MS SQL Server is now supported for reporting.
- Web reporting did not support MS SQL Server in R8.

P6 Extended Schema vs. ODS

With the introduction of the Enterprise Reporting Database in P6 Release 6.0, the ODS (Operational Data Store) introduced the ability to access types of P6 calculated data that were previously not directly accessible from the P6 database. The ODS is a separate database that is installed as part of the Enterprise Reporting Database. Once configured, customers typically are able to populate updates to the ODS nightly. While the ODS is useful for many reporting applications, some customers still require more up to date data access. The P6 Extended Schema addresses this issue by making storage of calculated data part of the main P6 database.

In P6 Release 8.0, native P6 reporting was done against the ODS. Starting in P6 Release 8.1, all P6 reporting will be done against the P6 Extended Schema. Customers will however still be able to report against the ODS database using an external reporting tool such as Oracle's BI Publisher. Refer to the Reporting section for more information on reporting.
Overview

In P6 Release 8.0, the P6 Reporting Database (formerly referred to as Enterprise Reporting Database) with its ODS database component was the primary source for report data. The timeliness of the data in the ODS database depended on when the Incremental Process was last run to update it, which for many customers is nightly. The Stage database, shown as follows, was used as an intermediate staging location for data calculation during the Incremental Process used to update the ODS and STAR databases. The STAR database is the data warehouse component of the P6 Reporting Database.

Note: The following figure reflects the structure of the P6 Reporting Database 2.0 (which is compatible with P6 Release 7.0) and the Reporting Database 2.1 (which is compatible with P6 Release R8).

Figure 1. Data Storage configuration in P6 Release 8.0
With the P6 Extended Schema implementation in Release 8.1, and the related Publication Services, described later in this document, the user has more control over what data is calculated and stored and the frequency at which this is done. The Stage database is no longer required, as its function has been replaced by the new Publication Services. The main purpose of the Publication Services is the calculation and storage of data into the P6 Extended Schema. In addition, the ODS and STAR databases are now populated directly from new logical views that are part of the P6 Extended Schema.

Figure 2. Data storage configuration in P6 Release 8.1
P6 Extended Schema

As previously mentioned, the P6 Extended Schema is the existing core P6 database schema, with the addition of new database tables to accommodate the storage of calculated data. Approximately 65 new tables have been added to the P6 database schema, as well as new logical views for reporting. The new tables are installed by default on upgrade to P6 Release 8.1. For more information on the new P6 Extended Schema refer to the P6 Release 8.1 schema documents.

A new database user, called PxRptUser, is also installed by default with P6 Release 8.1. This new database user supports reporting from the P6 Extended Schema. The PxRptUser has access to new logical views that are a combination of existing P6 database tables and the new P6 Extended Schema tables. This new user must be in the same instance as the P6 database for both Oracle and MS SQL Server installations. More details on how to use PxRptUser for reporting are covered in the Reporting section of this document.

Publication Services

There are two new services dedicated to publishing project data to the P6 Extended Schema tables:

- Project Arbiter service
- Publish Project service

The Project Arbiter service is used to automate the process of submitting Publish Project services for eligible projects to the service queue.

There are also four new services for publishing global data to the P6 Extended Schema tables:

- Publish Enterprise Data
- Publish Enterprise Summaries
- Publish Resource Management
- Publish Security

Each of these services, except for the Project Arbiter, will recalculate data and store it in physical fields in the extended schema tables, and will perform other data denormalization tasks. For example, duration fields are calculated fields in the existing P6 schema tables, but will be recalculated and stored as physical fields in the extended schema tables when a project is published.
The first time a project or global data type is published to the P6 Extended Schema tables, a full recalculation is performed against the project or global data, and all data is published to the extended schema tables. Once a project or global data type has been published, future publication services run against the same project or global data type will perform an incremental recalculation and publish only data that has changed since the last time it was published, and will recalculate the impact of those changes.

When any of the publication services are run, each business object type processed as part of the service is individually timestamped when completed, and this timestamp is stored in the database. Since the services are relying on individual business object type timestamps, and not a timestamp for the service as a whole, the services can accurately determine the data changes that have occurred to a business object since the last time it was processed.

Oracle recommends that the global data services be run before turning on the Project Arbiter service. In addition, the Enterprise Summaries service should be run after all the projects services have been completed.

**Publishing project data**

All Publish Project services must be submitted to the service queue by the Project Arbiter service. The Project Arbiter service runs at an interval you specify, and is intended to be run frequently (every minute by default). This ensures timely project data in the extended schema tables, and ensures that ASAP Publish Project requests are processed in a reasonable timeframe. Therefore, the Project Arbiter service must be enabled and configured properly in P6 in Administer > Application Settings > Services tab.

**Note:** The new Check Overallocation feature also requires the Project Arbiter service to be enabled because this feature will publish a project before running the Check Overallocation service.

The Publish Project service will recalculate and publish all project business objects containing calculated or denormalized data. This includes, but is not limited to, the WBS, activities, resource/role assignments, high-level planning assignments, relationships, risks, documents, expenses, steps, milestones, UDF values, notes, issues, budget change logs, timesheets, code assignments, funding source assignments, etc.

There are several ways a project can be queued for publication, and these methods vary depending on the state of projects in the database: initial state (no projects have been published) or steady state (all projects enabled for publication have been published).
**Initial state behavior**

In the EPS view, there is a new project-level setting called Enable Publication. It determines if a project is eligible to be published. On upgrade, this flag will be set to true for all projects. By default, the Project Arbiter service is disabled so you can decide which projects should be enabled for publication before you turn the Project Arbiter service on.

**Note:** All of the Application Settings mentioned in the remainder of this section are described in detail in [Configuring and Deploying Publication Services](#).

When you initially turn on the Project Arbiter service (‘Enable Publish Projects’), all projects enabled for publication will be available for queuing by the Project Arbiter service. During this initial state, the ‘Publish idle projects’ setting should be enabled if you want all projects enabled for publication to be published successively regardless of threshold settings for automatic publication.

Each time the Project Arbiter service runs during this initial phase, it will process the ‘Maximum number to publish’ you specify until all projects enabled for publication have been published once. The projects will be submitted to the queue based on descending Project ID. Depending on the number of projects in the database, the Project Arbiter service may need to run several times before all projects in the database will be added to the queue and published. Once all projects have been published during this initial phase, projects in the database are in a steady state. From this point forward, only projects actively being worked on will be published.

**Note:** You should not enable the ‘Publish idle projects’ setting if you only want projects actively being worked on published to the extended schema tables. If the ‘Publish idle projects’ setting is disabled, only the steady state behavior applies.

**Steady State Behavior**

Once the database is in a steady state, there are two methods for submitting Publish Project services into the queue:

- Manual submission (ASAP)
- Automatic submission based on thresholds

Regardless of which method is used to publish a project, a new field in the EPS view, Publication Priority, is used by the Project Arbiter service to determine the order in which it submits projects to the queue each time the arbiter runs.
Manual Submission (ASAP)
Any user with OBS access to a project or template can publish it from the EPS or Activities views, provided it is enabled for publication. In the EPS view, users can select any combination of projects or EPS nodes and click Actions > Publish Projects. In the Activities view, users can click Actions > Run > Publish Projects to publish the currently open projects or template. In either view, Publish Projects will be disabled only if none of the selected/open projects are enabled for publication.

When the user clicks Publish Projects, an internal date field (px_next_date) is set to the current timestamp of the database server for all selected/open projects enabled for publication. The next time the Project Arbiter service runs, it will read this date field and automatically submit a Publish Project service for the selected/open projects. Once the Project Arbiter has processed an ASAP request, users can view the status of the service in the Activities view (Actions > View Service Status).

Automatic Submission
A new table in the P6 schema, PLPROJREF, stores the timestamp of the last Publish Project service for each project, and tracks the number of changes made to the project in the PROJWBS, TASK, TASKRSRC, and TASKPRED tables since that timestamp.

Note: All of the Application Settings mentioned in the remainder of this section are described in detail in Configuring and Deploying Publication Services.

In the Application Settings > Services tab, you can specify a threshold for the number of changes made to a project (100 by default) that, when surpassed, will trigger the Project Arbiter to automatically submit a Publish Project service for a project. You can also specify a time threshold that will force an automatic publication (8 hours by default). Each time the Project Arbiter service runs, it will look to the PLPROJREF table to determine if the number of changes to a project has passed the threshold and, if so, submit the project to the queue. If the PLPROJREF table identifies at least one change to a project but the number of edits does not pass the threshold, the project will be queued if the time threshold has been surpassed.

When a new project is added in the steady state, the Enable Publication flag will be set to true and the project will be queued automatically when one of the threshold settings is passed. If you do not want new projects published before they contain a certain amount of detail, clear the Enable Publication flag on the project after you create it.

Finally, the following actions will cause a project to be automatically published the next time the Project Arbiter service runs (the threshold settings will be ignored):

- Changing the project baseline
- Changing the project calendar
- Changing the data date
- Enabling or disabling the ‘Calculate Activity % Complete from activity steps’ setting (EPS view > Actions menu > Set Project Preferences > Calculations tab)
• Changing the default price/unit for activities without assignments (EPS view > Actions menu > Set Project Preferences > Defaults tab > Price/Unit field)

Publishing Global Data
As previously mentioned, there are four new services that will publish global data to the P6 Extended Schema tables:

• Publish Enterprise Data
• Publish Enterprise Summaries
• Publish Resource Management
• Publish Security

Each of these services can be configured to run at regularly scheduled intervals, down to the hour. These recurring services can be defined in Administer > Global Scheduled Services. By default, these services are disabled on install/upgrade. Unlike the Scheduled Services introduced in R8, these services are predefined and can only be edited (not added/deleted). Only users with the new Administer Global Scheduled Services global privilege can edit these services.

Oracle recommends that the Publish Security service be run before any other global data service and before all Publish Project services. This will ensure that user security data is populated in the extended schema prior to reports being run against global and project data. Likewise, when defining the recurring intervals for the global data services, Oracle recommends the Publish Security service be configured to run before any other global data service so that security changes are applied to the extended schema as soon as possible. You can optionally run the Publish Security service manually after changing user security data to ensure that the extended schema security data is updated as soon as possible.

The Publish Enterprise Data service calculates and publishes enterprise-level business objects containing calculated and denormalized fields, including but not limited to cost accounts, currencies, calendars, codes, funding sources, resource teams, role teams, timesheet dates, and UDFs.

The Publish Enterprise Summaries service calculates and publishes EPS-level spread data for currently published projects in a given EPS.

The Publish Resource Management service calculates and publishes the resource hierarchy, resources, resource rates, resource limits, role rates, resource role assignments, and resource security.

The Publish Security service publishes users, the OBS, security profiles and calculates security for the P6 Extended Schema.
The P6 Extended Schema implementation and its related Publication Services provide customers with more flexibility and control over their data. Because the new database tables that comprise the P6 Extended Schema are installed by default on upgrade to P6 Release 8.1, there are no extra steps that need to be taken to install the new extended schema. However, the new Publication Services that populate the calculated data in the extended schema are turned off by default on upgrade and must be configured by an Administrator. The next sections describe the new Administrator settings and configuration details related to the Publication Services.

Also included is the recommended hardware configuration for a P6 Extended schema implementation and performance and scalability details.

**Configuring and Deploying Publication Services**

This section will define the order in which Publication Services should be configured, describe the available settings, and provide configuration best practices and considerations.

The configuration and deployment of Publication Services can be summarized in six steps:

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Define server-side service settings in the Administration</td>
</tr>
<tr>
<td>2</td>
<td>Define Publication Period settings</td>
</tr>
<tr>
<td>3</td>
<td>Set security for publication services</td>
</tr>
<tr>
<td>4</td>
<td>Configure and Deploy Global Scheduled Publication</td>
</tr>
<tr>
<td>5</td>
<td>Prepare projects for publication</td>
</tr>
<tr>
<td>6</td>
<td>Configure and Deploy the Project Arbiter Service</td>
</tr>
</tbody>
</table>

Each step is defined in the following corresponding section.

1. **Define server-side service settings in the Administration Application.**
The following table defines the new server-side settings for the publication services. These settings should be defined before defining application-level publication settings within P6.

<table>
<thead>
<tr>
<th>Setting Name</th>
<th>Default Value (recommended)</th>
<th>Valid Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Services/Publication/Settings/General Data/Page Size</td>
<td>32</td>
<td>1-8192</td>
</tr>
<tr>
<td>For general data business objects, the maximum number of business objects that can be processed per thread. General data refers to all business objects that are not time-distributed and not stored in a blob.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Services/Publication/Settings/General Data/Max Job Count</td>
<td>1</td>
<td>1-20</td>
</tr>
<tr>
<td>For general data business objects, the number of parallel threads that can be processing business objects. General data refers to all business objects that are not time-distributed and not stored in a blob.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Services/Publication/Settings/Time Distributed Data/Page Size</td>
<td>16</td>
<td>1-8192</td>
</tr>
<tr>
<td>For time-distributed business objects, the maximum number of business objects that can be processed per thread. Time distributed data is any business object with values stored in spread intervals (for example, assignment spreads).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Services/Publication/Settings/Time Distributed Data/Max Job Count</td>
<td>1</td>
<td>1-20</td>
</tr>
<tr>
<td>For time-distributed business objects, the number of parallel threads that can be processing business objects. Time distributed data is any business object with values stored in spread intervals (for example, assignment spreads).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Services/Publication/Settings/Blob Data/Page Size</td>
<td>25</td>
<td>1-8192</td>
</tr>
<tr>
<td>For blob data business objects, the maximum number of business objects that can be processed per thread. Blob data refers to all business objects that are stored in a blob field (for example, note fields).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Services/Publication/Settings/Blob Data/Max Job Count</td>
<td>1</td>
<td>1-20</td>
</tr>
<tr>
<td>For blob data business objects, the number of parallel</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Setting Name</td>
<td>Default Value (recommended)</td>
<td>Valid Range</td>
</tr>
<tr>
<td>-----------------------------------------------------------------------------</td>
<td>------------------------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>threads that can be processing business objects. Blob data refers to all business objects that are stored in a blob field (for example, note fields).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Services/Publication/Settings/Log/Severity Level</td>
<td>info</td>
<td>debug, info, warning, error</td>
</tr>
<tr>
<td>Log severity level for all publication services. The values are inclusive. For example, choose ‘debug’ to log all messages; choose ‘warning’ to log both warning and error level messages.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Services/Publication /Settings/Log/Global Service Log File Cap (Success)</td>
<td>5</td>
<td>1-25</td>
</tr>
<tr>
<td>For each global publication service, the maximum number of log files that will be stored for successful runs. Using Publish Enterprise Data as an example, if set to 5, the last 5 successful Publish Enterprise Data logs will be stored. When the limit is reached, the most recent successful run log will overwrite the oldest successful run log.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Services/Publication/Settings/Log/Global Service Log File Cap (Fail)</td>
<td>5</td>
<td>1-25</td>
</tr>
<tr>
<td>For each global publication service, the maximum number of log files that will be stored for failed runs. Using Publish Enterprise Data as an example, if set to 5, the last 5 failed Publish Enterprise Data logs will be stored. When the limit is reached, the most recent failed run log will overwrite the oldest failed run log.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Services/Publication/Settings/Log/Project Service Log File Cap (Success)</td>
<td>5</td>
<td>1-25</td>
</tr>
<tr>
<td>For the Publish Project service, the maximum number of log files that will be stored for successful runs, per project. Using Project A as an example, if set to 5, the last 5 successful Publish Project logs for Project A will be stored. When the limit is reached, the most recent successful run log will overwrite the oldest successful run log.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Services/Publication/Settings/Log/Project Service Log File Cap (Fail)</td>
<td>5</td>
<td>1-25</td>
</tr>
<tr>
<td>For the Publish Project service, the maximum number of log files that will be stored for failed runs, per project. Using Project A as an example, if set to 5, the last 5 failed Publish Project logs for Project A will be stored. When the limit is reached, the most recent failed run log will overwrite the oldest failed run log.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The following settings exist for each publication service. Rather than repeating for each service, they are described once here, and this description applies to each publication service.

<table>
<thead>
<tr>
<th>Setting Name</th>
<th>Default Value (recommended)</th>
<th>Valid Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Services/Publication/Services/&lt;serviceName&gt;/Interval</td>
<td>10s</td>
<td>1s-24d20h31m23s647</td>
</tr>
<tr>
<td>Amount of time to wait before running the next available job (in 1d1h1m1s format – specifying no letters implies milliseconds).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Services/Publication/Services/&lt;serviceName&gt;/Concurrent Threads</td>
<td>For Publish Project, default is 2 (all others 1)</td>
<td>For Publish Project, 0-20 All others, 0-1</td>
</tr>
<tr>
<td>The number of processes used for the service on this server. A value of 0 (zero) indicates that the service will not be run on this server.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Services/Publication/Services/&lt;serviceName&gt;/Active Mode</td>
<td>true</td>
<td>true/false</td>
</tr>
<tr>
<td>Process jobs continuously until all jobs are completed (true) or process each job according to Interval (false)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The following settings existed in previous versions of P6, and are listed here to provide recommended values for publication services.

<table>
<thead>
<tr>
<th>Setting Name</th>
<th>Default Value (recommended)</th>
<th>Valid Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thread Pool/Number of Threads</td>
<td>50</td>
<td>2-300</td>
</tr>
<tr>
<td>The number of server threads. Even though the minimum value is 2, the minimum value should be the number of enabled publication services plus one.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thread Pool/Maximum Long Running Task Duration</td>
<td>20m</td>
<td>&lt;duration&gt;</td>
</tr>
<tr>
<td>The maximum duration a thread can be used for one task.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Database/Instance/ConnectionPool[PML]/Renewable Leases</td>
<td>true</td>
<td>true/false</td>
</tr>
</tbody>
</table>
### Setting Name

<table>
<thead>
<tr>
<th>Setting Name</th>
<th>Default Value (recommended)</th>
<th>Valid Range</th>
</tr>
</thead>
</table>

If true, connection leases are renewed if database statements are completed within the MaxLeaseDuration time period. When true, the code can hold onto the connection as long as it needs, provided SQL statements are completed within the MaxLeaseDuration period. When true, the connection is revoked if no SQL statements are issued within the MaxLeaseDuration period or if one statement takes longer to execute than that period.

Used for the long running connection pool, which is used in the Business Rule Engine when scheduling long running jobs.

**Database/Instance/ConnectionPool[PML]/Maximum Lease Duration**

The maximum amount of time a database connection can be leased before it is revoked.

Used for the long running connection pool, which is used in the Business Rule Engine when scheduling long running jobs.

<table>
<thead>
<tr>
<th>Field</th>
<th>Default Value</th>
<th>Valid Range</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Start date</strong></td>
<td>First day of the previous year</td>
<td>NA</td>
</tr>
</tbody>
</table>
### Field

The date on which the first time period will begin for time-distributed data. Using 1/1/10 as an example, if the ‘Time distributed interval’ setting (see below) is set to Day, time-distributed data will be stored in daily intervals beginning on 1/1/10.

Regardless of the date entered as the start date, all time-distributed data in the database is still published to the extended schema tables. However, all time-distributed data that exists in intervals prior to the start date will be lumped into the first interval. For example, assume the earliest project start date in your database is 6/30/05, you set the ‘Start date’ to 1/1/08, and set the ‘Time distributed interval’ to Day. All time-distributed data that exists between 6/30/05 and 1/1/08 will be calculated and lumped into the first day interval, 1/1/08.

**Recommendation:** Set this value to the earliest project start date in the database if you want users to be able to produce time-distributed reports for any date range. If that is not a requirement, select a point in the past that represents a reasonable amount of historical spread data for your organization.

**Note:** This field can only be edited by Admin Superusers. Changing this value after data has been published to the extended schema will cause all project and global data to be automatically republished.

<table>
<thead>
<tr>
<th>Field</th>
<th>Default Value</th>
<th>Valid Range</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Finish date is current date plus</strong></td>
<td><strong>2 years</strong></td>
<td>1 month – 99 years</td>
</tr>
<tr>
<td>The period of time in the future, from the current date, for which you want time-distributed data calculated and stored in the extended schema tables. For example, assume today’s date is 6/30/11, and this value is set to 1 year. When a publication service is run on 6/30/11, the end date for time-distributed data will be 6/30/12. All time-distributed data that exists in the database after this calculated end date (in this case, 6/30/12) is still published to the extended schema tables but will be lumped into the last day or week interval ending on 6/30/12. <strong>Recommendation:</strong> Set this value to an interval that will allow users to produce time-distributed reports for a reasonable amount of time in the future. This value should typically be in the 2-5 year range. <strong>Note:</strong> This field can only be edited by Admin Superusers. Changing this value after data has been published to the extended schema will cause all project and global data to be automatically republished.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Time distributed interval</strong></td>
<td>Day</td>
<td>Day, Week</td>
</tr>
<tr>
<td>The interval in which publication services will calculate and store time-distributed data. <strong>Recommendation:</strong> Set to Day if this level of granularity is required for spread data. Set to Week if performance of the services is most important (this may only be necessary for very large databases). <strong>Note:</strong> This field can only be edited by Admin Superusers. Changing this value after data has been published to the extended schema will cause all</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Once your publication period settings are defined, you are ready to configure and deploy publication services.

Additional considerations when defining Publication Period settings

When specifying a date range be careful to not specify too broad of a date range - the broader the date range, the longer the duration for data calculations. The date range will determine the time frame for calculating project data, especially spread data, which takes the greatest duration of the P6 Extended Schema calculation. With an appropriate date range, spread data for the necessary time frame will be calculated and the rest of the data will be summarized and added to the beginning and end of the date range. Creating a date range that is too small will limit the amount of available data. When possible, follow the recommendations for each setting in the table above.

Time distributed data calculations are an intensive calculation and the Time distributed interval should be determined carefully. Changing this setting after setting up the Publication services would force a full recalculation of all data. This is not something that should be performed during regular production hours, but only during off time.

3. Set security for publication services

All users with OBS access to a project or template can publish it, but only privileged users can change a project’s Publication Priority. Likewise, only privileged users can edit global publication services.

The new Publication Priority project privilege is intended to be granted to administrator-level users who are not Admin Superusers, and not necessarily to users responsible for managing projects (these users might simply give highest priority to all of their projects). However, if used as intended, a project manager managing ten projects could rank their projects one through ten using the Publication Priority field so that the projects would be published in that order when all are published simultaneously. Regardless of who is assigned this privilege, the Publication Priority field can help control the flow of projects through the queue to ensure the most important projects get processed first.

To define these security settings for users:

1. In P6, click Administer > User Access.
2. Click Global Security Profiles in the left panel.
3. In the Global Security Profiles grid, select a profile that should have the ability to edit recurring global publication services.

4. In the Tools detail window, mark the Administer Global Scheduled Services checkbox.

5. Repeat for each global profile that should have this privilege enabled (this should be very few).

6. Click the Save icon.

7. Click Project Security Profiles in the left panel.

8. In the Project Security Profiles grid, select a profile that should have the ability to edit a project’s publication priority.


10. Repeat for each project profile that should have this privilege enabled.

11. Click the Save icon.

### 4. Configure and Deploy Global Scheduled Publication Services

Once the server-side service settings and publication periods are defined, you can configure recurring global publication services and start running them. Oracle recommends the global publication services be run prior to publishing projects.

To configure global scheduled services:

1. Log into P6 as a user with the global security privilege ‘Administer Global Scheduled Services’ or as an Admin Superuser.

2. Click Administer > Global Scheduled Services.

3. In the Global Scheduled Services dialog, select the first service in the list, Publish Security.

4. In the detail window, in the Run Service field, select how often you want the service to run.

The options below the Run Service field will change based on your selection. Also, in addition to the traditional recurring intervals (such as daily, weekly, and monthly), you can choose to run the service hourly, every three hours, every six hours, or every 12 hours. When making your choice, take into consideration the size of the database and server availability for running the services.

5. Enter or select a start time.
6. Repeat steps 4 and 5 for each global publication service.

   **Tip:** If you want to run each service in succession every time, in the Run Service field, select Run After Previous for each service except for the service listed first in the grid.

7. If using the Run After Previous option, select a service in the grid and use the up and down arrows in the toolbar to change the order. Be sure that the first service in the grid is not set to Run After Previous in the Run Service field.

   **Tip:** Oracle recommends that Publish Security be set to run prior to any other global publication service.

8. In the Enabled column, mark the checkbox for each global publication service.

   **Note:** Though the global publication services should be run prior to publishing projects, the Publish Enterprise Summaries service will not return meaningful results until all projects have been published. For this reason, you can optionally wait until projects have been published before enabling the Publish Enterprise Summaries service.

9. Click Save.

   Once your changes are saved, global publication services will begin running as scheduled (assuming the application servers running services are currently running). The first time each service is run, a full publication of all data related to that service will occur, which will take a significant amount of time depending on the size of the database.

5. Prepare projects for publication

As previously mentioned, there are new fields related to the Publish Project service that should be configured prior to enabling the Project Arbiter service, unless you intend to publish all projects in the database and the priority order of publication is not a consideration. These settings are:

- Enable Publication
- Publication Priority

An additional field, Last Published On, is also available for determining when a project was last published.

Prior to performing the steps in this section, determine what criteria you will use to identify whether a project is enabled for publication, and to specify a priority. For example, ask yourself questions like the following (answers will vary depending on the organization):
Should you exclude What If projects or templates from publication because your organization doesn't report against these types of projects?

Should you exclude projects completed more than 2, 3, or 5 years ago, or simply make them lower priority?

Should you exclude projects completed before the publication 'Start date' you specified in Application Settings?

By default, on upgrade, Enable Publication is set to true for all projects in the database, and the Publication Priority is set to 50 (1 highest priority, 100 lowest priority). To change these values:

1. In P6, click Projects.
2. If prompted to open a project, open any project (the project you select will not have any impact).
3. Click EPS in the Projects navigation bar.
4. On the EPS page, click Edit > Columns.
5. In the Customize Columns dialog, in the Project Publication grouping of Available Columns, move Enable Publication and Publication Priority to the Selected Columns list, then click OK.
6. To facilitate the editing of these fields, group or sort the view by a criteria you are using to determine which projects are enabled for publication and their priority (for example, project type or finish date).
7. Clear the Enable Publication checkbox for all projects you do not want to be published.
8. In the Publication Priority field, for each project enabled for publication, optionally enter a value between 1 (highest priority) and 100 (lowest priority).

**Tip:** You can use the Fill Down function in the Enable Publication and Publication Priority fields to quickly edit the values in these fields.
6. Configure and Deploy the Project Arbiter Service
As previously mentioned, the Project Arbiter service creates all Publish Project services. The Project Arbiter service can be enabled and configured in Application Settings. To configure and deploy the Project Arbiter:

1. Login to P6 as a user with the global privilege ‘Edit Application Settings’ or as an Admin Superuser.
2. Click Administer > Application Settings.
3. Click the Services tab.
4. In the Project Publication section, set values as appropriate. The following table describes each setting.

<table>
<thead>
<tr>
<th>Field</th>
<th>Default Value</th>
<th>Valid Range</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Enable Publish Projects</strong></td>
<td>False</td>
<td>True/False</td>
</tr>
<tr>
<td>Enables or disables the Project Arbiter service, which in turn determines if projects can be published. When the service is enabled, the fields below are available for editing.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Recommendation:</strong> This service should not be enabled until projects are ready for publication as described in the previous section, “Prepare projects for publication.”</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Publish projects every</strong></td>
<td>1 minute</td>
<td>1-99 minutes</td>
</tr>
<tr>
<td>The interval at which the Project Arbiter service will automatically run, from the time the service is turned on. Each time the Project Arbiter completes a run, it looks to this setting to see when it should run again; therefore, changing this setting will not require the application server to be restarted.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Recommendation:</strong> The interval should be set to a low number (less than 5 minutes) to ensure that ASAP Publish Project and Check Overallocation services are processed in a timely fashion. However, if your users will not be using these ASAP services, you can set this value higher.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Publish a changed project when the</strong></td>
<td>100</td>
<td>0-999</td>
</tr>
<tr>
<td><strong>Number of changes exceeds</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>For each project, the number of row changes among the PROJWBS, TASK, TASKPRED, and TASKRSRC tables that will force the Project Arbiter service to submit a Publish Project service for a project. A new table, PLPROJREF, tracks the changed rows for these tables since the last time the Project Arbiter ran. When the arbiter runs, it compares the number of changes for each project to the specified threshold, and submits a Publish Project service for each project that passes the threshold.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Recommendation:</strong> Set this value to a low number if you want to ensure that changes to projects are reflected in the extended schema tables as soon as possible. Set to a higher number if changes to a project do not</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>Default Value</td>
<td>Valid Range</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>need to be reflected in the extended schema immediately, or if the time threshold is sufficient in most cases for publishing projects.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Time since last publication exceeds</strong></td>
<td>8h</td>
<td>0-35791394h</td>
</tr>
<tr>
<td>For each project, the number of hours that must elapse since a project was last published before it will be automatically published again. When the Project Arbiter runs, if the PLPROJREF table has tracked at least one change for a project since the last time the Project Arbiter ran, it will submit a Publish Project job for a project if the time threshold has been surpassed.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Recommendation</strong>: Set this value to a timeframe in which your project data must be current in the extended schema tables. For example, if you set this to 24h, this ensures that all projects actively being worked on will be published at least once a day, even if the edit threshold is not passed.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Publish idle projects</strong></td>
<td>False</td>
<td>True/False</td>
</tr>
<tr>
<td>Mark this checkbox to ignore the threshold settings and publish in succession all projects enabled for publication until all projects have been published once. This setting only applies during the initial state, after upgrade, when no projects have been published. Once all projects that are enabled for publication have been published once, projects will only be added to the queue based on the threshold settings.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Recommendation</strong>: Do not mark this checkbox if you only want to publish projects actively being worked on to the extended schema tables. If your organization does not report against completed projects, it may not be necessary to publish projects not actively being worked on.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Maximum number to publish</strong></td>
<td>100</td>
<td>0-999</td>
</tr>
<tr>
<td>The maximum number of idle projects that can be added to the queue by the Project Arbiter service each time it runs. During the initial state where all projects are being published for the first time (all projects are idle), the Project Arbiter looks to this setting to determine how many projects to add to the queue each time it runs. Once all projects have been published and the database is in a steady state, this setting no longer applies.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong>: If you choose to deploy P6 to your organization before the database is in a steady state and all projects have been published once (not recommended), this setting will behave as follows: Each time the Project Arbiter service runs, it submits Publish Projects services for all projects it determines must be published because a user requested it or because a project passed a threshold. After submitting these projects to the queue, it then determines how many Publish Project services are currently in the queue, and subtracts that number from the ‘Maximum number to publish’ value to determine how many idle projects it can submit to the queue. If the number of projects currently in the queue</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>Default Value</td>
<td>Valid Range</td>
</tr>
<tr>
<td>-------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>exceeds the 'Maximum number to publish,' the arbiter will not add any additional idle projects to the queue.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Recommended Hardware Configuration**

**ARCHITECTURE OVERVIEW**

The new Publication Services are built into our Oracle Primavera P6 Enterprise Project Portfolio Management product which is a Java 2 Platform, Enterprise Edition (J2EE platform) web application. The division of tiers allows the application to scale according to customer’s performance demands.

The main tiers of Oracle Primavera P6 Enterprise Project Portfolio Management are:

- **The presentation tier** – A web server layer rendering JSPs, JavaScript, Applets, etc., to present a feature-rich UI accessible through various supported browsers. This tier allows the end users to schedule and view job status of the Publication Services.

- **The middle tier** – A J2EE application server forms the middle tier where all business logic of Oracle Publication Services is implemented. This is a scalable layer of the configuration that can be scaled horizontally or vertically. The customer can control the number of services that run on each server by working with multiple P6 Configuration groups.

- **The data tier** – The data tier consists of a standalone or clustered RDBMS environment utilizing Java Database Connectivity (JDBC) to integrate with the middle tier.

Publication Services reside on an application server (one or more) and the repository of the application data resides on the database server. The figure 1 illustrates the architecture of Oracle Primavera P6 suite of Products.
Performance and Scalability Considerations

While there are multiple ways to achieve required performance and scalability levels in Oracle Primavera P6 Enterprise Project Portfolio Management, the performance considerations can be grouped in two categories: Vertical and Horizontal. There are several pros (and cons) of choosing one over the other. Organizations can decide based on the required level of performance, availability requirements, short-term vs. long-term outlook of system usage, seasonality and frequently used application areas as well as freshness of the needed reports.

**Vertical Scalling (Scaling Up)**
Vertical scaling involves adding additional resources or upgrading resources on an existing system. Vertical scaling is usually a good approach if the application bottlenecks are processor and memory-related.

**JVM Heap Size**
The application objects such as Projects, Activities, and Assignments, etc., are stored in the Java Virtual Machine (JVM) heap allocation. Most of these objects are short lived, and are periodically cleaned up by the JVM’s garbage collection mechanism. As the number of concurrent Publication Service Jobs increases, performance and scalability are affected by the available heap space in the JVM. Increasing the heap size is an easier way to achieve the required performance and scalability.
Hardware Upgrade
Required performance and scalability can also be achieved by upgrading CPU, adding extra cores, upgrading to faster I/O devices, and upgrading from a 32-bit to 64-bit hardware (we recommend a 64-bit system to start).

Operating System Upgrade
Another way to achieve the required performance level is by upgrading to latest versions of the operating system, installing latest patch updates and by upgrading from 32-bit version to a 64-bit version, etc.

While vertical scaling is easier to achieve, it does not address the availability requirements. If the required level of availability is high, then vertical scaling alone will not be sufficient.

Horizontal Scaling (Scaling Out)
As the demand for application grows, additional nodes can be added to an existing application server cluster to handle the increased system load. For high availability requirements, horizontal scaling is the better option.

Adding Application Server Nodes
As the usage of application grows within the organization adding additional server nodes is the best way to achieve the necessary performance and scalability. If the organization’s business model exhibits seasonality or periodic variations, the system load will fluctuate accordingly. For example, the average load on the system might greatly vary during month end closing, or when the plant is closed for a week every quarter for maintenance. Consider adding or removing application server nodes to manage seasonality. To mitigate risk of degraded performance and undesired downtime, it is crucial to understand the business cycles of the organization and plan for the optimal level of performance, availability and scalability.

Database Scaling and Clustering
Database server scaling options are available, and have been widely adopted and implemented. Database clustering enables multiple nodes in a clustered system to mount and open a single database that resides on shared disk storage. This configuration provides high availability in the database environment.

Oracle Primavera Publication Service Server
The Publication Service process is integrated within the P6 application server for platform independence, better performance and scalability. Publication Services are responsible for executing real-time and scheduled application jobs.
The following application areas are processed as jobs:

**P6 – Services**
- Summarizer
- Scheduler
- Leveler

**Publication Services**
- Enterprise Data
- Enterprise Sum
- Resource Management
- Security
- Project
- Arbiter

The services are capable of processing large number of projects, activities, resources, calendars and resource assignments. Numbers of concurrent jobs largely affect CPU and memory requirements of the application server and load on the database servers.

- For all deployments, it is recommended that a dedicated application server node be set up for Publication Services. This application server should not be part of the cluster that processes HTTP requests from the web client. Horizontally scaling out by adding more dedicated job service application server nodes can address increased performance requirements.

- For the initial Publication Services run, off-peak hours or before going live are recommended if upgrading from previous versions. For example, scheduling a job to run when the load on the system is low or over the weekend implementation.

**What can Affect the Services**
The newly designed Services are highly scalable and robust. The following factors could impact the response time of all Services:

- Number of Activities/Assignments
- Length of project
- Depth of WBS Hierarchy
- Length of Activities/Assignments
- Number of Financial Periods
- Date Range
- Number of Projects
• Number of threads dedicated to the services
• Page size for the threads

DEPLOYMENT CATEGORIES
Oracle Primavera P6 Extended Schema deployments can be classified into three categories i.e. small, medium and large. Some of the factors considered for defining these categories are outlined below. These factors influence the hardware and software specifications during Oracle Primavera P6 Extended Schema deployment.

<table>
<thead>
<tr>
<th>Number of Objects</th>
<th>Deployment Categories</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Small</td>
</tr>
<tr>
<td>Projects</td>
<td>200</td>
</tr>
<tr>
<td>Active Users</td>
<td>50</td>
</tr>
<tr>
<td>Activities</td>
<td>100,000</td>
</tr>
<tr>
<td>Activities per project</td>
<td>5,000</td>
</tr>
<tr>
<td>Resources</td>
<td>500</td>
</tr>
<tr>
<td>Resource Assignments</td>
<td>100,000</td>
</tr>
<tr>
<td>Resource Assignments per project</td>
<td>5,000</td>
</tr>
<tr>
<td>Risks</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 1 – Deployment Categories

DEPLOYMENT ARCHITECTURES

Small Deployment – Single Node
This deployment model is suitable for a business unit or a division within an organization. It also is useful for setting up a pilot with the intent of moving to a medium or large size deployment. This deployment can achieve the required performance or scalability, but does not address the high availability requirement due to single point of failure.

Application Server Configuration

| CPU                  | Intel Xeon 5000 series (Quad Core 3.46 GHz) or equivalent |
Java Heap Size | 2 GB
---|---
Operating System | Oracle Enterprise Linux (OEL) 64-bit or Windows7 server 64-bit or equivalent

**Service Host Configuration**

<table>
<thead>
<tr>
<th>CPU</th>
<th>Intel Xeon 5000 series (Quad Core 3.46 GHz) or equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java Heap Size</td>
<td>4 GB</td>
</tr>
<tr>
<td>Operating System</td>
<td>Oracle Enterprise Linux (OEL) 64-bit or Windows Server 64-bit or equivalent</td>
</tr>
</tbody>
</table>

**Database Server Configuration**

<table>
<thead>
<tr>
<th>CPU</th>
<th>Intel Xeon 7000 series (Quad Core 2.66 GHz) or equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAM</td>
<td>2 GB</td>
</tr>
<tr>
<td>Operating System</td>
<td>Oracle Enterprise Linux (OEL) 64-bit or Windows7 server 64-bit or equivalent</td>
</tr>
</tbody>
</table>

**Medium Deployment – Clustered**

This deployment model is suitable for a full organization. This deployment should achieve the required performance or scalability; if performance is not where it is needed, this deployment can cluster another Publication Service host to reach required needs.

**Application Server Configuration**

<table>
<thead>
<tr>
<th>CPU</th>
<th>Intel Xeon 5000 series (Quad Core 3.46 GHz) or equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java Heap Size</td>
<td>4 GB per node</td>
</tr>
<tr>
<td>Operating System</td>
<td>Oracle Enterprise Linux (OEL) 64-bit or Windows Server 64-bit or equivalent</td>
</tr>
</tbody>
</table>

**Service Host Server**

<table>
<thead>
<tr>
<th>CPU</th>
<th>Intel Xeon 5000 series (Quad Core 3.46 GHz) or equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java Heap Size</td>
<td>6 GB</td>
</tr>
<tr>
<td>Operating System</td>
<td>Oracle Enterprise Linux (OEL) 64-bit or Windows Server 64-bit or equivalent</td>
</tr>
</tbody>
</table>

**Database Server Configuration**

<table>
<thead>
<tr>
<th>CPU</th>
<th>Intel Xeon 7000 series (Quad Core 2.66 GHz) or equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAM</td>
<td>4 GB per node</td>
</tr>
</tbody>
</table>
**Large Deployment – Clustered**

Assuming that high availability is required for a large deployment, the application server is clustered. A large deployment involves a high system load due to large data sets, processing, concurrent users etc. To handle this load, it is recommended to add a clustered database server, such as Oracle RAC Database. Due to the intense computations typically seen at large deployments, a large JVM heap is highly recommended. Horizontally scaling out by adding more nodes can address increased performance requirements.

**Application Server Configuration**

<table>
<thead>
<tr>
<th><strong>CPU</strong></th>
<th>2 Intel Xeon 5000 series (Quad Core 3.46 GHz) or equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Java Heap Size</strong></td>
<td>8 GB per node</td>
</tr>
<tr>
<td><strong>Operating System</strong></td>
<td>Oracle Enterprise Linux (OEL) 64-bit or Windows7 server 64-bit or equivalent</td>
</tr>
</tbody>
</table>

**Service Host Server**

<table>
<thead>
<tr>
<th><strong>CPU</strong></th>
<th>Intel Xeon 5000 series (Quad Core 3.46 GHz) or equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Java Heap Size</strong></td>
<td>8 GB per node</td>
</tr>
<tr>
<td><strong>Operating System</strong></td>
<td>Oracle Enterprise Linux (OEL) 64-bit or Windows7 server 64-bit or equivalent</td>
</tr>
</tbody>
</table>

**Database Server Configuration**

<table>
<thead>
<tr>
<th><strong>CPU</strong></th>
<th>2 Intel Xeon 7000 series (Quad Core 2.66 GHz) or equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RAM</strong></td>
<td>8 GB per node</td>
</tr>
<tr>
<td><strong>Operating System</strong></td>
<td>Oracle Enterprise Linux (OEL) 64-bit or Windows7 server 64-bit or equivalent</td>
</tr>
</tbody>
</table>

**OTHER FACTORS**

This document covers the performance of the overall Architecture of the Publication Services. However, the factors around the database setup play a very important role on performance. The following factors could impact the database performance:

- Hardware architecture and OS
- NIC (number of NICs, speed and duplex settings)
- Number of database instances on a server (dedicated vs. shared)
- Disk storage system performance (I/O speed, Buffer, Mirroring)
- Tablespace layout and extent sizing
- Table data, index, and lob distributions on tablespaces
- Table and index fill factor definition
- Database block sizing
- Connection management (dedicated vs. MTS)
- RAM allocations (automatic, SGA, PGA, Shared Pool, Buffer Pool, etc.)
- CBO optimizer parameter configuration setting
- Database table and index statistics gathering mechanism and frequency
- Anti-virus software
- Additional database jobs

For additional performance and scalability information related to the P6 Extended Schema implementation please refer to the P6 8.1 Sizing Guide.
In P6 Release 8.0, the primary source of data for reports generated from P6 Web was the ODS database (part of the P6 Reporting Database). As a result, report data was only as timely as the last time the incremental process ran to update the ODS. In P6 Release 8.1, the primary source of data for reports generated from P6 will be the new P6 Extended Schema, which will provide near real time reporting. The P6 Enterprise Reporting Database is no longer required for P6 reporting.

In P6 Release 8.1, the creation and administration of reports is done within Oracle’s BI Publisher tool as it was in P6 Release 8.0. A few key things have changed:

- P6 Release 8.1 now supports integration with BI Publisher 10G and 11G. Only BI Publisher 10G was supported in P6 Release 8.0.
- P6 Release 8.1 also supports reporting against Oracle or MS SQL Server databases. Only Oracle database installations were supported in P6 Release 8.0. The ODS database will still only support Oracle installations.
- When running reports from within P6 Web, the P6 Extended Schema is now the primary source for report data.
- The P6 Web reporting security model is different in P6 Release 8.1 when using the P6 Extended Schema as the source for report data. Refer to the Security section below.

**Report Security**

In P6 Release 8.1, the security model for reporting against the P6 Extended Schema starts with the new PxRptUser database user. PxRptUser is added automatically on upgrade to P6 Release 8.1 for Oracle and MS SQL Server database installations.

PxRptUser has full access to the logical reporting views that exist in the P6 Extended Schema that are a combination of existing P6 database tables and the new extended schema tables. These new logical views include three security tables that map each P6 user’s security access to global, resource and project data. A data source must be configured in BI Publisher to connect to the P6 Extended Schema using the PxRptUser account. When a report is launched from P6 Web or BI Publisher, BI Publisher must first make the connection to the P6 Extended Schema with PxRptUser. The user logged into either application is then passed as a security context to the P6 Extended Schema to apply security to the data set being returned in the BI Publisher report.

**Note:** In P6 Release 8.1, a user must have the “Edit EPS Costs/Financials” project privilege to be able to see project costs in a report generated from the P6 Extended Schema.
The P6 Extended Schema implementation requires that a security context be created in BI Publisher to establish security for P6 reports. Depending on your database installation it will be done in one of two ways:

1. For Oracle database installations:
   a. If you are using Oracle BI Publisher 10G, at minimum you must have the BI Publisher patch10623023 installed.
   b. Your P6 related data source in BI Publisher must point to the P6 EPPM database (using the new PxRPTUser database user).
   c. The ‘Use Proxy Authentication’ option must be unchecked in this data source definition.
   d. The ‘Pre Process Function’ for the P6 data source (JDBC Connection) in BI Publisher must be updated with the following string: Secpac.SET_BIP_USER(:xdo_user_name).

2. For MS SQL Server installations:
   a. Your P6 related data source in BI Publisher must point to the P6 EPPM database (using the new PxRPTUser database user).
   b. The ‘Use Proxy Authentication’ option must be unchecked in this data source definition.
   c. You must add the security context to each individual report. For more information see the “Creating Reports” section later in this document.

Upgrade

The new P6 Extended Schema logical views use the same naming convention that was used in the logical views in the ODS database. This means that reports previously run against the ODS database will run against the new P6 Extended Schema. However, on upgrade to P6 Release 8.1 your Oracle BI Publisher installation will need the following modifications in order for your existing P6 reports to run against the P6 Extended Schema:

1. If you are using Oracle BI Publisher 10G, at minimum you must have the BI Publisher patch10623023 installed.
2. Your P6 related data source in BI Publisher (previously configured to point to the ODS database) must be changed to point to the P6 EPPM database (using the new PxRPTUser database user).
3. The ‘Use Proxy Authentication’ option must be unchecked in this data source definition.
**Note:** In P6 Release 8.0, an ODS database user was created to mirror each P6 user with module access to Enterprise Reporting. These ODS users also included P6 security, so a proxy connection was needed to enforce P6 security when running a BI Publisher report. The duplication of each P6 user with Enterprise Reporting module access in the ODS database easily supported database authentication. In P6 Release 8.1, the ODS is no longer the primary source for P6 Reporting. Therefore, proxy authentication is no longer a valid option, and the ideal authentication model is LDAP or Custom. If administrators wanted to use a database authentication model, they would have to manually create database users.

4. For an Oracle installation, the ‘Pre Process Function’ for the P6 data source (JDBC Connection) in BI Publisher must be updated with the following string: Secpac.SET_BIP_USER(:xdo_user_name).

BI publisher reports can still run against the ODS when launching the report directly from within BI Publisher (a separate P6 Reporting Database license is required), assuming they have not been updated to use a P6 EPPM 8.1 data source as described in Step 2 in the preceding procedure. When launching a BI Publisher report from within P6 Web Release 8.1, reports configured to connect to the ODS will no longer be supported. P6 Web Release 8.1 will only support BI Publisher reports configured to connect to the new P6 Extended Schema.

**Creating Reports**

All report creation must be done in BI Publisher. It is recommended however, that once the reports are tested and pushed out for general use that the reports be executed through P6 only.

**Recommended steps for creating reports:**

1. It is recommended that report creation and testing be done in a Testing environment, as opposed to a Production environment. Once the reports are created and tested in BI Publisher, they can then be deployed for use in a P6 Production environment.

2. The report creator will log into BI Publisher as the new PxRPTUser that has been introduced as part of the P6 Extended Schema implementation in P6 Release 8.1.

3. A data source that connects to a P6 Extended Schema database must be selected for the reports. The username for the data source must be the PxRPTUser. For information on how to correctly configure a data source to connect to a P6 Extended Schema database refer to the Report Security section.

4. Once the report is created, the report should be tested to make sure that the appropriate data is returned. The PxRPTUser has access to all published P6 Extended Schema data so the report will return the correct information.
5. **For Oracle database installations ONLY**: The administrator can then log out of BI Publisher and log back in as a user with restricted P6 Extended Schema access to verify that security is being applied to the report.

6. **For MS SQL Server database installation ONLY**: After the report is tested for accuracy, additional steps must be taken to add security to the report.

   a. A `p_security_context_id` parameter must be added to the parameter list of each report.
      
      Note: P6 uses the `p_security_context_id` parameter to pass the User ID of the P6 User executing a BI Publisher report from P6 to the P6 Extended Schema `rpt_user_id` field. This ensures that the user executing a given report only sees the data for which they have security access.

   b. The `p_security_context_id` parameter must be applied, using a WHERE clause, to each Select statement in a report that accesses P6 Extended Schema tables that require security. For example:
      
      ```sql
      select id from project WHERE project.rpt_user_id = :p_security_context_id ORDER by id
      ```

   c. Once the `p_security_context_id` parameter and WHERE clauses are added, the report creator can test the report by setting the default value for the `p_security_context_id = 0` for a P6 Admin Superuser. To test the report with limited security access, specify the default value for the `p_security_context_id` to a P6 User ID with the appropriate security access.

   d. **Tip:**
      
      i. The `where` clause is only useful when running a report in BI Publisher against an MS SQL Server P6 Extended Schema data source when the `p_security_context_id` parameter is populated with a valid default value. If the `p_security_context_id` parameter is not populated with a valid default value, P6 security will not be enforced when running a report in BI Publisher against an MS SQL Server P6 Extended Schema data source.
      
      ii. If the WHERE clause is not included, or is incorrect in the report definition, P6 Web will return an empty dataset on report execution.
      
      iii. `WHERE <table>.rpt_user_id = 0`
      
      • This must be added to all views that would contain a security control (such as all project related data, codes, etc.) when testing P6 reports in BI Publisher as an Admin Superuser.
      
      iv. `WHERE <table>.rpt_user_id = null`
      
      • For the ActivityCode and ActivityCodeType tables there is no security around global codes, so the WHERE clause must also include: where rpt_user_id is null.
v. Refer to the sample reports included with P6 Release 8.1 for examples on how to create BI Publisher reports for use with P6 and an MS SQL Server database.

For more information on configuring P6 Release 8.1 for reporting refer to the P6 EPPM Administrators Guide. For more detail on administering reports in BI Publisher refer to the BI Publisher documentation.
Eventing

If you have eventing configured for P6, and have enabled the events for the new Publication Services, the system will publish events to the JMS destination when the jobs of the following Publication Services are finished in 'Completed' or 'Failed' status:

- Publish Project
- Publish Enterprise Data
- Publish Enterprise Summaries
- Publish Resource Management
- Publish Security

The event messages contain the status of the job (either 'Completed' or 'Failed') and the timestamp when the job was finished. For the Publish Project service, the message contains information for the project (ObjectId, Id and Name).

Note: The Project Arbiter service does not trigger events.
P6 Extended Schema FAQ

Question #1: Will the reports I created to run against the ODS run against the P6 Extended Schema once I have upgraded to P6 Release 8.1?

Answer #1: Yes they will, with some minor modification. Refer to the Reporting Upgrade section of this document for more details.

Question #2: Who can publish projects? Who can publish global data?

Answer #2: Any user with OBS access to a project or template can publish it, provided the project or template is enabled for publication. Any user with the new global privilege ‘Administer Global Scheduled Services’ can enable, disable, or edit the recurring interval for a global publication service.

Question #3: Can all types of projects be published?

Answer #3: No. As a general rule, any project or template you can see in the EPS view can be published, including checked out and what-if projects. Types of projects you cannot see in the EPS view (such as baselines, scenarios, and requested projects) cannot be published.

Question #4: Can I change application-level publication settings after data has already been published?

Answer #4: Yes. In Application Settings > Services tab, you can change all Project Arbiter settings (Project Publication section) at any time. The Project Arbiter reads the state of these settings each time it completes a run so that it knows what the values are set to for the next run. In the Publication Periods section, you can change these settings at any time. However, changing any of these settings will force a full recalculation of all projects and all global data the next time each publication service runs.

Question #5: If the timeliness of the report data available in the P6 Extended Schema depends on the Publication Services, how likely is it that there may be an inconsistency between the live data in P6 and the calculated data stored in the logical reporting views of the P6 Extended Schema?
**Answer #5:** P6 Release 8.1 is designed to support “near real time” reporting. Thus, a data mismatch can occur for short periods of time. A user might publish a project in order to run a report against that project. While the publish process is occurring, another user might update the live data that would impact a calculation stored in the extended schema. This would create a data mismatch between the live P6 data and the stored calculation when the report is run. If the arbiter is set to run frequently (every minute or so), and the number of edits exceeds the threshold that will trigger an automatic publication, the project will be published again rather quickly. That would resolve the mismatch. Of course, any time a user sees a mismatch in a report, the user can republish the project again manually to correct the issue.

**Note:** The speed at which a project is republished is dependent upon how busy your system is (other projects waiting to be republished), and on how many servers are dedicated to the Publication Services.

**Question #6:** How do I make my Publish Project service run faster?

**Answer #6:** There are three ways to increase the speed at which the Publish Project service executes:

1. Consider dedicating a server to the Publish Project service. Refer to the P6 EPPM Administrator’s Guide for information on this.
2. Review the recommendations for the Publish Project settings available in the Administration Application covered in the Define server-side service settings in the Administration Application section of this document.
3. Review the recommendations for the Project Arbiter settings available in the Configure and Deploy the Project Arbiter Service section of this document.

**Question #7:** What do I do when there is a global or project publication service failure?

**Answer #7:** If there is a failure in either the Publish project or Publish global data services, the administrator can refer to the Publication Services log files for more detail.

The Publication Services log files can be found in the BREBootstrap <logDir>\services folder on the application server. In the ‘\services’ folder there will be a sub-folder for each Publication service type. For example, there will be a ‘jt_project’ (jt stands for Job Type) folder for Publish Project job service runs. Other sub-folders include: jt_enterprisedata, jt_enterprisesum, jt_resourcemgmt, jt_security.

Publication Service log files are suffixed with the overall status of the job service run. For example, a successful run will be saved as ‘03.30.2011-10.28.18_SUCCEEDED.log’ and a failed service run will be logged as ‘03.30.2011-10.28.18_FAILED.log’. You can review the contents of a '_FAILED.log' and search for occurrences of the text: ‘SEVERE’ as this is the header/prefix for error messages in the log file. The error messages should help you determine what type of business object data and
potentially the record(s) that are causing the failure. You can then correct the data in the P6 applications and then re-run the services. If you are seeing errors along the lines of InterruptedException, OutOfMemoryError and/or Connection/Session/Lease timeouts it may be necessary to adjust settings found in the P6 admin configuration under Services>Publication>Settings (refer to the admin guide for more information).

It may be necessary to increase the level of logging detail written to the log file in cases of FAILED job service runs. This can be found in the P6 Administrator application under Services>Publication>Settings>Log. Increase the log level to ‘debug’ to include more detailed troubleshooting information into the log file.

If there is a failure in any of the global or project publication services, report content could possibly not reflect the current state of the system.

**Question #8**: Even though ODS was not always up to date, users running reports at least knew it was up to date as of a certain date/time, which was always the same. How can I achieve the same confidence that report data is accurate as of a certain date/time if projects and global data are always being published?

**Answer #8**: There are three options:

**Option 1**: Establish a process that sets aside a window of time to run all of the Publication Services during off-peak hours. For example, if you operate 9-5, you can have all of your Publication Services set to run every night @ midnight, and have BI reports generated/emailed @ 6:00am.

**Option 2**: If you do not have an off-peak hours window, then your data is as close to near-time as your process defines the intervals for the Publication Services to execute. If you run your services every hour and they take less than an hour to execute (in the steady state; after initial state is complete) then you can say your reports are accurate to within the past hour, etc.

**Option 3**: There is a RPT_CURRENT_FLAG on the Activity, ResourceAssignment and WBS logical report views which tells you on a row by row basis what items are current and which are not.

**Question #9**: How do I clear a published project’s data, whose publication I have turned off, from the P6 Extended Schema if I no longer want that project’s data included in reports?

**Answer #9**: We do not currently have a process in place to remove a specific project from the P6 Extended Schema logical reporting views once it has been published. A workaround could be to make use of portfolios to filter out published project(s) that you no longer wish to report on.

**Question #10**: How do I clear a deleted project’s data from the P6 Extended Schema if I no longer want that project’s data included in reports?
**Answer #10:** Use the ‘clean_px_delete’ script found in the admuser schema for Oracle, or in the dbo schema for MS SQL Server.

- To execute this script for Oracle: `exec clean_px_delete;
- To execute this script in SQL Server: `clean_px_delete
  go`

This will remove all data in the Extended Schema tables that no longer have associated records in live tables.

**Question #11:** Can I force a full recalculation of a project to update the P6 Extended Schema data?

**Answer #11:** After the initial publication of a project, all changes thereafter will be incremental. There is, however, a back door way to trigger a full recalculation of a project after it has already been published. To force a full calculation on a project null out the field PX_SAFETY_DATE on the project table in the P6 EPPM database.

  For example: `update project set px_safety_date = null where px_enable_publication_flag = 'Y'

**Question #12:** How does the P6 Extended Schema implementation impact the ODS and STAR databases that are part of the P6 Reporting Database?

**Answer #12:** The ODS and Star schemas will now be populated by the Publication Services and the logical views that are part of the P6 Extended Schema implementation. The P6 Extended Schema settings under Application Settings \ Services \ Publication Period will be used for the ODS and Star reporting databases for the time-distributed interval and the data range of dates to be published. The ‘Time Distributed Interval’ setting of Day or Week will represent the spread data bucket type to be used in ODS and Star. The ‘Start Date’ and ‘Finish date is current date plus’ settings will represent the range of data to be used in the data warehouse.