

Oracle
Primavera
Analytics Planning and Sizing Guide for On-Premises

Version 20
December 2020

Contents

Overview of Planning Your Implementation.....	5
About Oracle Primavera Data Warehouse	5
About Oracle Primavera Analytics	6
About Oracle Primavera P6 Enterprise Project Portfolio Management	7
About Oracle Data Integrator (ODI)	7
About Oracle Primavera Unifier	8
Working with Oracle Primavera Analytics	9
Critical Performance Factors	10
Pulling Data between Servers	10
Merging Updates into Target Database	10
PL/SQL-based Transformations	11
Planning Process	11
Requirements Phase.....	11
About Partitioning.....	11
Significance of Operational Reporting.....	12
Planning Phase.....	13
Full ETL Process Only	13
Monitoring Usage	13
Basic Monitoring.....	14
Physical Sizing	14
Table Partition for Multiple Data Source and History Data.....	15
Archiving and Restoring History	16
Archiving History	16
Restoring Archived History	17
Physical Components.....	17
ETL Process Server.....	18
Calculating Spread Sizes	18
Queries for Spread Sizing	18
Estimating the Size of the Primavera Data Warehouse Database	19
Primavera Data Warehouse Database	20
Oracle Data Integrator (ODI) Sizing.....	20
ODI Sizing for P6 EPPM Projects	20
Primavera Data Warehouse Database for ODI	21
ODI Master and Work Repository	22
ODI Java EE Agent	22
Physical Hardware.....	22
Monitoring Dimension and History Level Impacts on the Database.....	23
Consideration for ETL Performance	23

Network.....	24
Memory	24
Extract and Load.....	25
Oracle Data Integrator (ODI)	25
Summary of Physical Hardware Sizing.....	25
Size of Database	25
Amount of Change.....	25
Complexity of Project Data.....	26
Expectations and Operations.....	26
Database Servers.....	27
Planning Revisited.....	27
Timing Estimation.....	27
Factors Affecting ETL Runtime	29
Planning for the Impact of P6 EPPM on Primavera Data Warehouse	31
Choosing Whether to Use Partitioning	31
Effect of the P6 Extended Schema on Primavera Data Warehouse	32
Date Range.....	32
Enable Publication	33
History Settings on Project Level.....	33
Slowly Changing Dimensions.....	33
Multiple Primavera Data Warehouse Environments.....	34
Creating Multiple Data Sources for a Single Primavera Data Warehouse Database	34
Setting up Multiple Data Sources	35
Filtering by Data Source in Analytics.....	35
Single Primavera Data Warehouse, Single Data Source, Multiple ETLs.....	36
Appendix A: Checklists for Primavera Analytics and Primavera Data Warehouse.....	36
Pre-installation for an Oracle Platform.....	36
Pre-installation for a Microsoft SQL Server Platform	37
Pre-installation for Configuring the ETL Server.....	37
Pre-installation for Oracle Data Integrator (ODI)	38
Pre-installation for the Primavera Data Warehouse Database	39
Installation: Installer And Configuration.....	40
Installation: Running The Processes.....	42
Post-installation: After The Processes Complete.....	42
Appendix B: Database Scripts For Validations.....	43
Copyright.....	45

Overview of Planning Your Implementation

Primavera Data Warehouse and Analytics are data warehousing products. Implementing them involves databases, servers, and controlling an ETL process. Oracle Primavera provides the code to perform the ETL process. The ETL process can be run from PL/SQL database or Oracle Data Integrator (ODI) for on-premises. Analytics contains a set of pre-built dashboards, analyses, and underlying metadata to provide end users with easy access to information.

When planning each deployment, you need to consider:

- ▶ What is required for physical storage and CPU processing.
- ▶ Which components of Primavera Data Warehouse and Analytics will be implemented.
- ▶ The differences in data volumes, frequency of data changes, and business requirements.

To successfully implement these products:

- 1) Review the critical performance factors.
- 2) Outline a methodology for planning an installation.
- 3) Determine the physical storage requirements of the data warehouse.
- 4) Address the server performance requirements of the STARETL process.

In This Section

About Oracle Primavera Data Warehouse	5
About Oracle Primavera Analytics	6
About Oracle Primavera P6 Enterprise Project Portfolio Management.....	7
About Oracle Data Integrator (ODI)	7
About Oracle Primavera Unifier	8

About Oracle Primavera Data Warehouse

Primavera Data Warehouse works with the P6 EPPM database and the Primavera Unifier database to provide a robust and powerful reporting solution.

Primavera Data Warehouse enables an organization to perform advanced business analysis on project and portfolio data. It supplies a dimensional schema that organizes P6 EPPM and Primavera Unifier hierarchical relationships. The Primavera Data Warehouse schema allows organizations to store data for History Levels (Project, WBS, Activity, Business Process, Cost Sheet, Cash Flow, Assignment, User-Defined Fields, or Codes) for a specific History Interval (Year, Quarter, Financial Period, Month, Week, or Day). This data allows for tracking trends for advanced business intelligence. Primavera Data Warehouse captures data for activities, resource assignments and utilization, space management, business processes, cash flows, cost sheets, and user-defined fields. Primavera Analytics requires Primavera Data Warehouse.

About Oracle Primavera Analytics

Oracle Primavera Analytics provides an in-depth and comprehensive method for analyzing and evaluating facilities and real estate management, project performance, project history, resource assignments and utilization, business processes, cash flows, and cost sheets.

Built upon the Oracle Business Intelligence (OBI) suite, Primavera Analytics delivers a catalog of analyses that provide an interactive way of viewing, analyzing, and evaluating P6 EPPM and Primavera Unifier data. In addition, Primavera Analytics provides a repository definition that contains the data mappings between the physical data and the presentation layer of OBI.

The dashboards provide detailed insight into your P6 EPPM and Primavera Unifier data through analytical charts, tables, maps, and graphics. Dashboards allow you to navigate to other analyses to provide precise root-cause analysis. OBI allows you to configure individual analyses with P6 EPPM and Primavera Unifier Action Links, enabling you to navigate directly to your P6 and Unifier site for true "Insight to Action" capabilities. You can save an analysis created with OBI in the OBI Presentation Catalog and integrate the analysis into any OBI dashboard. You can enhance results through options such as charting, results layout, calculations, and drill-down features.

Use Primavera Analytics to:

- ▶ Perform root-cause analysis and employ management-by-exception.
- ▶ Gather critical insights into current and historical performance of all projects, programs, and portfolios.
- ▶ Drill down from aggregated metrics to examine the root-cause of a problem.
- ▶ Make better decisions to eliminate project failure.
- ▶ Quickly visualize critical project performance in early stages.
- ▶ Identify and predict cost sheet and cash flow trends early in the project life cycle.
- ▶ Gain visibility into resource performance through s-curve analysis.
- ▶ Show staffing needs by portfolio with early warning indicators for upcoming under-staffed project work.
- ▶ Roll-up business processes to understand trends and trends by various dimensions.
- ▶ Use geospatial visualization to view project, activity, and resource data metrics by geographic location with full drill-down capabilities.

Primavera Analytics provides a repository definition to use with the OBI suite. The repository definition contains:

- ▶ A physical representation of Primavera Data Warehouse.
- ▶ A business layer to perform customized calculations.
- ▶ A presentation layer that groups all the calculated business layer fields into logical subject areas.

The repository definition delivers an extensive list of key performance indicators (KPIs) from Primavera Unifier, and P6 EPPM. Depending on the data source, this list includes (but is not limited to) Earned Value, Costs, Units, Percent Completes, Counts, Business Processes, Lean, Resource Assignment, Schedule Health, Cash Flow, and Cost Sheets. It also enables data to be sliced by various dimensions, including time, EPS, portfolios, projects, activities, resources, project hierarchies, cost breakdown structures, and business processes.

Primavera Analytics delivers a sample dataset, consisting of Primavera Data Warehouse data, where the dashboards and analyses in the catalog were built. You can use this sample data to view the power of dashboard and analyses delivered in the catalog, and see how you can integrate the catalog with your data.

About Oracle Primavera P6 Enterprise Project Portfolio Management

Oracle Primavera P6 Enterprise Project Portfolio Management (P6 EPPM) is a robust and user-friendly integrated solution for globally prioritizing, planning, managing, and executing projects, programs, and portfolios. It optimizes role-specific functionality to satisfy each team member's needs, responsibilities, and skills. It provides a single solution for managing projects of any size, adapts to various levels of complexity within a project, and intelligently scales to meet the needs of various roles, functions, or skill levels in your organization and on your project team.

Thousands of companies rely on P6 EPPM to:

- ▶ Plan, schedule, and manage the right strategic mix of projects
- ▶ Make informed project, cost, and resource management decisions
- ▶ Enhance team collaboration and workflow productivity
- ▶ Maximize each project contributor's skills and responsibilities
- ▶ Track progress and optimize capacity for maximum profitability
- ▶ Deliver programs and projects on time and within budget
- ▶ Share data with human capital, financial management, and enterprise resource planning systems

About Oracle Data Integrator (ODI)

Oracle Data Integrator (ODI) Enterprise Edition delivers high performance data movement and transformation among enterprise platforms with its open and integrated E-LT architecture and extended support for Big Data. Oracle Data Integrator Enterprise Edition is critical to leveraging data integration initiatives on-premise or in the cloud, such as Big Data management, Service Oriented Architecture and Business Intelligence. An easy to use user interface combined with a rich extensible framework helps Oracle Data Integrator Enterprise Edition improve productivity, reduce development costs and lower total cost of ownership among data centric architectures. Oracle Data Integrator Enterprise Edition is fully integrated with Oracle Fusion Middleware, Oracle GoldenGate, Oracle Database, Oracle Big Data Appliance and Exadata to put data at the center of your enterprise. Oracle Data Integrator Enterprise Edition is open and standards based to work with 3rd party applications as well as Oracle's applications.

For On-Premises

Use either the PL/SQL-based or ODI-based StarETL process.

Note: The term, StarETL, when used generically, applies to PL/SQL as well as ODI.

About Oracle Primavera Unifier

Designing, building, and managing facilities requires extensive collaboration between numerous, often geographically dispersed, disciplines and entities. Throughout the process, from conceptual design to facility operations, access to accurate, up-to-date information is critical to the success of a project and facility.

Oracle Primavera Unifier is an integrated platform that optimizes business processes and creates visibility to enable customers to better manage all of the communications and information required to successfully manage a facility throughout the life cycle.

Primavera Unifier is a system for managing the flow of information in projects or shells, providing a seamlessly automated and integrated environment across the life cycle of your company's facilities, from planning, design, procurement, construction and into operations and maintenance. It provides real-time visibility across multiple projects or shells to help your company make fast, accurate decisions.

Primavera Unifier lets you track and manage information such as budgets, project or shell members, specifications, requests for information, and shared documents. You decide who has access to the information, which team members are allowed to approve changes to the information, and how information flows between people.

Oracle Primavera's solutions automate manual processes and pull together information from various point systems typically used on a portfolio of projects or shells. Through Primavera Unifier, executives and project or shell team members can better manage all data and business processes in one centralized system, while reducing the reliance on older technologies such as e-mail, fax, and desktop applications.

Primavera Unifier was designed from the ground up specifically for the facility owner, based upon our industry domain expertise and knowledge of best practices combined with direct customer input gathered over decades of client interaction. The result is a robust set of capabilities with an intuitive, easy-to-use interface. Primavera Unifier enables leading owners and operators to increase enterprise efficiencies, reduce project and operating costs, enhance visibility, and improve time-to-market.

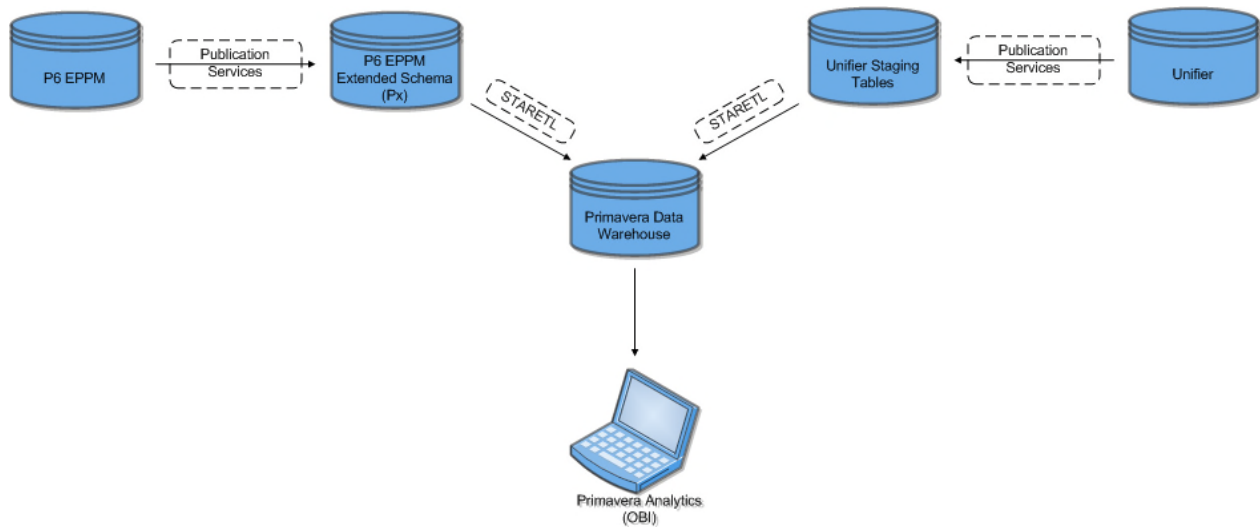
Working with Oracle Primavera Analytics

Depending on your organization's specific deployment, P6 EPPM and Primavera Unifier work together with Primavera Data Warehouse and Primavera Analytics as depicted below.

For On-Premises

Use either the PL/SQL-based or ODI-based StarETL process.

Note: The term, StarETL, when used generically, applies to PL/SQL as well as ODI.



Applications:

P6 EPPM and Primavera Unifier: P6 EPPM and Primavera Unifier are the primary interfaces for administering and managing projects. In one or both of these applications, you will gather the data on which Primavera Analytics will report. P6 EPPM and Primavera Unifier provide an extensive array of features designed to optimize all phases of Project Management, Resource Management, and Portfolio Management. They include full support for activities, work breakdown structures, planning and scheduling, costs, resource administration and assignment, codes, user-defined fields, roles, teams, and capacity planning.

Oracle Business Intelligence (OBI): OBI integrates data from multiple enterprise sources, and transforms it into key insights that enable strategic decision-making, drive continual business process improvements, and promote alignment across the enterprise. OBI can also generate proactive alerts based on integrated reports data mined from Primavera Data Warehouse.

Primavera Analytics: This integrated dynamic reporting tool uses the gathered data to facilitate the creation of ad hoc reports and interactive custom dashboards reflecting trends and metrics for activities, portfolios, resource assignments, utilization, and project history.

Technology:

The P6 EPPM Database: The main database for P6 EPPM data. The P6 EPPM database includes the P6 Extended Schema (Px) optimized for reporting.

The Unifier Database: The main database for Primavera Unifier data. The Primavera Unifier database includes the Primavera Unifier Staging Tables optimized for data extraction.

Primavera Data Warehouse: Primavera Data Warehouse consists of the Star database. It pulls data from the P6 EPPM Extended Schema and the Primavera Unifier Staging Tables, which can then be used to create reports. The Primavera Data Warehouse database accumulates project and cash flow data over time, providing baselines for tracking trends and for advanced business intelligence; it also provides key metrics on current data, which allows you to drill through root cause analysis. You can use tools, such as OBI, to analyze the data in Primavera Data Warehouse.

Critical Performance Factors

The ETL process has four areas that will affect performance:

- 1) Pulling data between servers.
- 2) Managing the updates of the component databases.
- 3) Performing PL/SQL and direct SQL transformation operations on the database server.
- 4) Project management data dimension and fact table generations.

Pulling Data between Servers

Data movement revolves around the E (Extract), T (Transform), and L (Load) in ETL. In a typical implementation, you will deploy the database (either P6 EPPM or Primavera Unifier) and data warehouse on separate physical servers. Because of this, you must maximize the bandwidth and minimize latency between servers. Ideally, the servers reside in the same data center with gigabit Ethernet connecting the servers. (Oracle performance tests servers in a central data center with gigabit connections.) You should verify throughput of server communication by performing basic file copy or FTP operations between the servers.

The data movement process is based on the standard SQL, either with INSERT-SELECT or CREATE TABLE AS (CTAS) syntax using an Oracle database link. While the Oracle RDBMS is efficient at moving data through the link, the overall performance depends on the physical network connection.

Merging Updates into Target Database

Primavera Data Warehouse leverages PL/SQL bulk operations to process smaller batches while performing interim commits. This results in linear scaling of update operations. For example, if it takes five minutes to update 1,000,000 rows, it will take ten minutes to update 2,000,000 rows). Rows per second performance remains constant regardless of amount of change or data size.

Many of the processes, including the PL/SQL bulk operation processes, are run in parallel threads. The use of parallel PL/SQL processes not only increases the potential scalability of this process, but also the demand for CPU powers and network bandwidth on the server.

PL/SQL-based Transformations

PL/SQL performs a portion of the transformation process. The largest portion of this is referred to as direct SQL. SQL update statements run directly against tables to perform simple transformation that does not require business logic. These processes are generally run in parallel on the database server (and are very CPU intensive). With the large chunk of data processed, the disk input/output (IO) performance is limiting. A high-performing storage area network (SAN) file system improves the overall system response time.

Planning Process

The planning process includes a Requirements Phase (where you evaluate the needs of users) and a Planning Phase (where you evaluate impacts on your system).

Requirements Phase

The first phase of the planning process is the Requirements phase, where you understand what the needs of users. This includes:

- ▶ The types of reports users want
- ▶ The level of report detail users need
- ▶ The time frame users need reports delivered
- ▶ How current the data should be

Primavera Data Warehouse provides a wide variety of results. However, you may not need them all for a given installation. There are two categories of reporting solutions:

- ▶ Operational
- ▶ Analytics/Business Intelligence

About Partitioning

Oracle Database Enterprise Edition with Partitioning is the solution for growth over time within the history tables. While Oracle recommends that you use a partitioned environment, Primavera Data Warehouse also supports non-partitioned environments.

Note: If you are connecting multiple P6 EPPM or Primavera Unifier data sources to Primavera Data Warehouse, you must enable partitioning.

If you do not use a partitioned environment, Oracle recommends that the P6 EPPM database and/or Primavera Unifier database be a "small-sized database" as defined in *Planning for the Impact of P6 EPPM on Primavera Data Warehouse* (on page 31). Partitioning helps keep performance consistent over time for each STARETL process run. This is particularly important when enabling activity or WBS-level history. Without partitioning, you should not use activity and WBS-level history gathering.

Without partitioning, the amount of time required to run the STARETL process increases over time. As the size and row count of specific Primavera Data Warehouse tables increases (w_project_history_f, w_wbs_history_f, w_activity_history_f, as well as tables ending in _HF and _HD) the need for partitioning also increases. You can monitor the run times for each STARETL process with the logs and reports located in **<PDW Install Folder>\star\etl_homes\staretl<id>\log**. See the *Primavera Data Warehouse Installation and Configuration Guide* for details on log files.

If you deploy without partitioning, the following features are not supported:

- ▶ WBS-level or activity-level history transfer to Primavera Data Warehouse
- ▶ WBS-level or assignment-level history transfer to Primavera Data Warehouse
- ▶ Multiple data sources
- ▶ Burn Down functionality
- ▶ Work Planning functionality
- ▶ Capturing dimensional history

Note: If you implement partitioning, you must license the Oracle Database partitioning option.

Significance of Operational Reporting

Operational reporting in P6 EPPM covers the day-to-day reports used by project managers, resource managers, and other tactical personnel. When using operational reporting, you must consider the scheduling and delivery of reports. The combination of the spread data persistence in the P6 EPPM schema and the reporting tool (BI Publisher) addresses the scheduling, execution, and delivery of the reports. On a day-to-day basis, the reporting load will be fairly consistent. You need to consider the following:

- ▶ **When will reports run?** Users may need to execute on demand or they may need reports prior to the start of the work day. These considerations will affect the timeliness of the data. You can run a report one or more times per day and limit the duration of each run (which is proportional to the data volume and date range selected by the users).
- ▶ **How will reports be delivered?** You must get the right reports to users at the right time. BI Publisher offers multiple ways to deliver reports from the P6 EPPM Extended Schema. These include email, HTTP, WEBDAV, and direct printing. You must consider the logistics of setting up these delivery methods during the planning process. Moreover, there are leading third-party reporting tools that can be utilized to generate and deliver the reports.

Note: The BI Server component is capable of robust caching of query results, which can mitigate performance concerns. The effectiveness of caching depends on how much users share security. If project access of every user is distinct (including the level of access to cost fields) then the cache will only be effective for each user individually.

Planning Phase

This implementation will place high demands on physical storage. Because the calculation process places a unique demand on the ETL, this data warehouse implementation may require higher amounts of CPU/memory. Subsequent sections of this document will provide more detailed information. First, this document will examine the high-level aspects of planning the implementation.

Full ETL Process Only

This release runs only in full ETL mode. Since the data has been pre-processed by the publication services in the P6 EPPM Extended Schema and in the Primavera Unifier Staging Tables, direct loads of the data into Primavera Data Warehouse are more efficient.

Monitoring Usage

You must understand how data compiles in key tables and how this impacts performance of the usage monitoring process. You can gather audit columns on each table (for example, CREATE_DATE) on a daily basis and from the REFRDEL table. To make hardware decisions, you must monitor the size of the tables and how much the tables grow over several weeks. In addition, you must track the peak usage times.

For P6 EPPM, this includes the following tables:

- ▶ PROJECT
- ▶ TASK
- ▶ TASKACTV
- ▶ TASKRSRC
- ▶ TASKMEMO
- ▶ UDFVALUE
- ▶ RSRCHOUR
- ▶ PROJWBS

For Primavera Unifier, this includes the following tables:

- ▶ SYS_STG_BP_LINE
- ▶ SYS_STG_CASH_FLOW
- ▶ SYS_STG_CBS_HIER
- ▶ SYS_STG_CBS_ROW
- ▶ SYS_STG_PROJECT

Basic Monitoring

Use the following query to count changes to the table (insert and update):

```
SELECT count(*)
FROM <table>
WHERE update_date > trunc(sysdate);
```

Use the following query to read/select rows from REFRDEL in a single step:

```
SELECT TABLE_NAME, count(*)
FROM REFRDEL
WHERE delete_date > (sysdate);

GROUP BY TABLE_NAME
ORDER BY TABLE_NAME
```

When needed, you can use P6 Auditing to get a precise picture of usage.

Notes:

- These queries should run at the end of the day to capture all the changes from that day.
 - These queries must be repeated for all the critical tables.
-

Physical Sizing

The physical space requirements of the data warehouse consist of more than just copies of the project management data. Space requirements will vary with the amount of data processed from the P6 EPPM database. The system uses space for the following types of data:

- ▶ **Core Project Management Data** includes all the physical fields that exist in the P6 EPPM database. This is approximately all the data in the P6 EPPM database.
- ▶ **Logical Fields** become physically stored as part of the P6 EPPM database in the Extended Schema (Px) tablespace. While this is less than the size of the P6 EPPM Database, it may be as much as 50% of the total space.
- ▶ **Fact/Spread Data** total size depends on the number of activities and resource assignments, the average length of activities, and the total window (date range) of the data warehouse. Because of this, it will be treated as a distinct group. It is fundamental to the dimensional schema (Primavera Data Warehouse).
- ▶ **Other ETL Tables** use database space specific to the ETL process. This space is trivial relative to the core P6 EPPM data.
- ▶ **ETL Process Installation** includes shell scripts, SQL files, and JAR files.

Table Partition for Multiple Data Source and History Data

- ▶ **Purpose of Table Partition** Improved performance and ease of data management.
- ▶ **Multiple Data Source** Primavera Data Warehouse supports multiple P6 and Primavera Unifier databases as the data source.

By default, the related tables are value partitioned based on the DATASOURCE_ID. The Partitioned tables are visible by viewing `\scripts\create_star_tables_part.sql` and searching for tables with **PARTITION**.

- ▶ **History Data** For Primavera Data Warehouse, you can control history interval and levels settings on the project level in P6 or Primavera Unifier. History data can grow quickly. For example, if you select Activity as the History Level, this automatically sets the History Interval for Activity and Resource Assignment history to Daily, which can affect the ETL process performance.

By default, these tables are interval-partitioned based on the PERIOD_END_DATE value, then sub-partitioned by the value of the DATASOURCE_ID. Oracle recommends you allocate of separate tablespace for each partition.

The P6 EPPM history tables are as follows:

- ▶ **Project history**
W_PROJECT_HISTORY_F
- ▶ **WBS history**
W_WBS_HISTORY_F
- ▶ **Activity history**
W_ACTIVITY_HISTORY_F

The Primavera Unifier history tables are as follows:

- ▶ **Business Process history**
W_BP_HIST_F
- ▶ **Cash Flow history**
W_CASH_FLOW_HIST_F
- ▶ **Cost Sheet history**
W_COST_SHEET_HIST_F

The indexes on these tables are LOCAL indexes per each partition.

- ▶ **Slowly Changing Data for projects with Activity as the History Level** This automatically sets the History Interval for Activity and Resource Assignment history to Daily, which can affect the ETL process performance. By default, these tables are interval partitioned based on the effective_end_date value, then sub-partitioned by the value of the DATASOURCE_ID. The tables related to this feature end in `_HD` and `_HF`.
- ▶ **Number of Partitions** are dynamically determined for the interval partition. Ten data source partitions are created by default. The following is an example of how to add more data source partitions:

- ▶ **Value Partition Addition**

```
alter table W_ACTIVITY_D add partition P11 values (11) tablespace
star_hst1;
```

- ▶ **Interval Partition Management** The merging of partitions may be applicable in some environments if the record counts found in each partition are relatively small. The **<INSTALL_HOME>/scripts** folder contains a `history_partition_management.sql` script. This script has examples on how to merge partitions based on an upper thresholds and a function to drop empty partitions (that may have been created in error). This script serves as a template in partition management.

Archiving and Restoring History

This section describes how to archive and restore archived history from the database.

Archiving History

To archive history, you need to take the history partition and move the data out of the database into a file.

To archive history:

Note: The commands below represent the first history partition.

- 1) Drop security on the `W_ACTIVITY_HISTORY_F` so you can exchange the partition to a table. Run the following command to drop security:

```
begin
  dbms_rls.drop_policy('STARUSER', 'W_ACTIVITY_HISTORY_F',
    'W_ACTIVITY_HISTORY_F_P_POLICY');
  dbms_rls.drop_policy('STARUSER', 'W_ACTIVITY_HISTORY_F',
    'W_ACTIVITY_HISTORY_F_PC_POLICY');
  commit;
end;
/
```

- 2) Exchange the partition into a new table by running this command:

```
create table r1_p1 as select * from staruser.w_activity_history_f where 0=1;
alter table staruser.w_activity_history_f exchange subpartition r1_p1 with table
r1_p1;
```

- 3) Export the table or move it to another database. If you are using Oracle Database, see the Data Pump documentation for your version of Oracle for details.
- 4) Remove the table from the database:

```
drop table r1_p1;
```

- 5) Enter one of the following commands to restore security by running the STARETL process:

- ▶ With Windows:

```
setup.bat 44 44
```


- ▶ With UNIX or Linux:

```
setup.sh -from 44 -to 44
```

Restoring Archived History

To restore archived history to the database:

- 1) Drop security on the W_ACTIVITY_HISTORY_F by running this command:

```
begin
  dbms_rls.drop_policy('STARUSER', 'W_ACTIVITY_HISTORY_F',
'W_ACTIVITY_HISTORY_F_P_POLICY');
  dbms_rls.drop_policy('STARUSER', 'W_ACTIVITY_HISTORY_F',
'W_ACTIVITY_HISTORY_F_PC_POLICY');
  commit;
end;
/
```

- 2) Import the table into the STARUSER database. If you are using Oracle Database, see the Data Pump documentation for your version of Oracle for details.
- 3) Restore the archived table to the STARUSER database.
- 4) Restore the partition by exchanging the archive table into the correct partition. To restore the partition run this command:

```
alter table w_activity_history_f exchange subpartition r1_p1 with table r1_p1;
```

- 5) Remove the table by running this command:

```
drop table r1_p1;
```

- 6) Enter one of the following commands to restore security by running the STARETL process:

- ▶ With Windows:

```
setup.bat 37 37
```

- ▶ With UNIX or Linux:

```
setup.sh -from 37 -to 37
```

Physical Components

You should consider two physical components for sizing the data warehouse:

- ▶ ETL Process Server
- ▶ Primavera Data Warehouse Database

These physical components will be treated as separate instances of the Oracle database or a physical server, although this is not required. There is no direct size impact on the database beyond normal space usage, and it is not considered as one of the physical components to be sized.

ETL Process Server

While this server is the central controller of the ETL process, it represents only a small portion of physical space used. The only files, other than the ETL process files, are log files from each run and the Temporary Flat files for xlat (language translations), which are minimal.

Calculating Spread Sizes

The actual number of spread sizes depends on factors such as calendar work days, large differences in dates relative to the data date, ETL time duration, and the exclusion of zero value rows. Estimation size for SPREAD fact tables is more than 3 GB, which are all a part of the total tablespaces sizes.

A quick calculation uses a "best guess" on the average number of days for activities and resource assignments.

Total Activities: 1,000,000 X Average Activity Length: 5 = Total Spreads: 5,000,000

Queries for Spread Sizing

Activity Spread Estimate Based on Median Activity Length

```
select
median(
greatest(nvl(target_end_date,to_date('12122000','mmddyyyy'))
,nvl(act_end_date,to_date('12122000','mmddyyyy'))
,nvl(reend_date,to_date('12122000','mmddyyyy'))
,nvl(rem_late_end_date,to_date('12122000','mmddyyyy')) )
-
least(nvl(target_start_date,to_date('12122199','mmddyyyy'))
,nvl(act_start_date,to_date('12122199','mmddyyyy'))
,nvl(restart_date,to_date('12122199','mmddyyyy'))
,nvl(rem_late_start_date,to_date('12122199','mmddyyyy')) )
) * count(*) Spread_Rows
from task t inner join project p on p.proj_id = t.proj_id and orig_proj_id is null
and p.px_enable_publication_flag='Y'
where task_type in ('TT_Task','TT_Rsrc');
```

Resource Assignment Spread Estimate Based on Median Activity Length

```
select
median(
greatest(nvl(tr.target_end_date,to_date('12122000','mmddyyyy'))
,nvl(tr.act_end_date,to_date('12122000','mmddyyyy'))
```

```

,nvl(tr.reend_date,to_date('12122000','mmddyyyy'))
,nvl(tr.rem_late_end_date,to_date('12122000','mmddyyyy')) )
-
least(nvl(tr.target_start_date,to_date('12122199','mmddyyyy'))
,nvl(tr.act_start_date,to_date('12122199','mmddyyyy'))
,nvl(tr.restart_date,to_date('12122199','mmddyyyy'))
,nvl(tr.rem_late_start_date,to_date('12122199','mmddyyyy')) )
) * count(*) Spread_Rows
from taskrsrc tr inner join project p on p.proj_id = tr.proj_id and orig_proj_id
is null and p.px_enable_publication_flag='Y'
inner join task t on t.task_id = tr.task_id
where task_type in ('TT_Task','TT_Rsrc');

```

Estimating the Size of the Primavera Data Warehouse Database

Only the fact tables will be considered for Primavera Data Warehouse database sizing purposes because they are responsible for most of the data. Spread and resource limit data are initially loaded into holding tables (_FS suffix), so sizes are doubled for these tables.

Data Component	Calculation	Rows	Size Example
W_ACTIVITYSPREAD_F	300 bytes/row	5,000,000 x 2	3.0 GB
W_RESOURCE_ASSIGNMENT_SPREAD_F	175 bytes/row	5,000,000 x 2	1.6 GB
W_RESOURCE_LIMIT_F	70 bytes/row	1,825,000 x 2	0.125 GB
Dimensional and Temporary	20% of Spread	n/a	0.9 GB
W_Project_History_F	265 bytes/row	6000	0.04 GB
Total			5.6 GB

Note: The above sizing does not include the W_Activity_History_F and W_WBS_History_F tables, which will be the largest if the History Interval is set at Weekly and the History Level is set at Activity.

Primavera Data Warehouse Database

The Primavera Data Warehouse database contains a dimensional data model along with several fact tables (including UDF fact tables and history fact tables, if configured) and the supporting dimensions. Excluding the configurable history data tables, the Primavera Data Warehouse database will contain fewer rows because baseline projects are not directly accessible.

The fact data represents the largest portion of data in the Primavera Data Warehouse database. As with any Primavera Data Warehouse schema, this data is the most granular data built to support roll-up queries.

Depending on the History Interval and History Level settings, the Activity, WBS history tables, and Slowly Changing Dimension (_HD) tables are likely to be the largest.

The primary two fact tables contain activity and resource assignment spread data respectively.

The next largest fact table contains resource utilization data (W_RESOURCE_LIMIT_F). This differs from other fact tables in that the data size is not a function of the number or size of projects. Instead, it is a function of the number of resources in the database, and the size of the data warehouse reporting window. There is a daily value for everyday of the reporting period, and for each resource. For example, if the reporting window spans five years (1,825 days), and there are 1,000 resources in the database, the total recorded in the fact table will be 1,825,000.

The Project History fact table is the smallest and it has only project-level data. The difference is that this table is a trending table with snapshots of the data over time. The number of snapshots depends on the interval chosen during installation (weekly, monthly, financial period). The granularity of this fact table goes to only the project-level; it contains no spread information. Calculate the number of rows using the total non-baseline projects times the number of snapshots. This will grow over time, so the yearly total for a 10,000 project database with weekly snapshots will be 520,000 rows.

Oracle Data Integrator (ODI) Sizing

This section provides ODI sizing estimates for P6 and Primavera Data Warehouse.

ODI Sizing for P6 EPPM Projects

Primavera Analytics supports immediate project ETL for a selected set of P6 projects, while in the database, PROJECT.ETL_INTERVAL=0. Any updates to tables that belong to these projects, such as, Activity, Resource Assignment, etc., triggers the immediate ETL. Due to the duration of ETL, usually a few minutes for an update, ensure that a minimal set of projects are selected. Otherwise, the performance of immediate ETL will deteriorate.

For commodity hardware, select a maximum of 10 projects. The number of projects depends on the activity and resource assignment spread rows.

As a P6 database user, run the following query which returns the number of spread rows for projects enabled immediate ETL in P6 database:

```
select sum(total_rows) from (
```

```

select count(*) total_rows from activityspread where projectobjectid in
(select proj_id from project where etl_interval = 0 and project_flag='Y'
and px_enable_publication_flag='Y' and orig_proj_id is null)

union

select count(*) total_rows from resourceassignmentspread where
projectobjectid in (select proj_id from project where etl_interval = 0 and
project_flag='Y' and px_enable_publication_flag='Y' and orig_proj_id is
null)

);

```

It is recommended that the number of rows should be approximately 10,000.

Primavera Data Warehouse Database for ODI

The sizing requirements are the same the traditional star ETL process.

It is recommended to create all tablespaces as BIGFILE TABLESPACE

- ▶ STAR_DAT1
- ▶ STAR_HST1
- ▶ ODI_TGT_DAT1
- ▶ UNDOTBS1

Additionally, you will need to:

- ▶ UNDO tablespace size for staruser.
This may require unusual large size if spread data is huge. Usually it needs to hold UNDO for both activity and assignment spread whole tables inserts.
- ▶ Set ODI_TGT_USER tablespace size to hold all data for all data sources.

Tablespace	Files	Size
UNDOTBS1	undotbs01.dbf	32G
ODI_TGT_DAT1	ODI_TGT_DAT1.dbf	32G
	ODI_TGT_DAT2.dbf	17G

- ▶ Set STAR_DAT1 and STAR_HST1 tablespace.
- ▶ Ensure you plan ahead for adding extra data files, especially for STAR_HST1.
- ▶ Plan Processes in pfile or spfile
120 concurrent connections per P6 data source if all load plans run at the same time
 - ▶ Concurrent connection needed by load plans:
 - On Demand: 20
 - Real time project: 20
 - Real time global: 30
 - Daily: 30
 - Load single project: 20
 - ▶ 30 concurrent connections per Unifier data source

For optimal values see *Doc ID 240710.1*

https://support.oracle.com/epmos/faces/DocumentDisplay?_afLoop=485391318785332&id=240710.1&_afWindowMode=0&_adf.ctrl-state=98tnne430_4 to decide after the ETL is stable for a period of time.

ODI Master and Work Repository

Determine the tablespace of work repository to store the running logs for x number of days. The default is 7 days.

Add at least 4 or 5 additional data files to the default tablespace DEV_ODI_USER. If you use rcu tool to create the ODI related schema, add a second data file to this tablespace as follows:

```
alter tablespace DEV_ODI_USER add datafile 'DEV_odi_user_2.dbf' size 100m
autoextend on maxsize unlimited;
```

ODI Java EE Agent

Use the following estimates for sizing:

- ▶ Connection pool size for ODI master repository: ~15 per data source
- ▶ Connection pool size for ODI work repository: ~5 per data source
- ▶ Go to ODI java EE agent weblogic server Services\Data Sources\<<datasourceName>\Connection Pool, Recommended values for ODI master and work repositories:
 - ▶ odiMasterRepository
 - ▶ Initial Capacity: 10
 - ▶ Max Capacity: 150
 - ▶ min Capacity: 10
 - ▶ odiWorkRepository
 - ▶ Initial Capacity: 10
 - ▶ Max Capacity: 150
 - ▶ Min capacity: 10

After running the java EE agent for a period of time, to decide the optimum values, go to:java EE agent weblogic server Services\Data Sources\<<datasourceName>\Monitoring and use **Active Connections High Count** as a reference to set the optimal value.

Physical Hardware

When evaluating the physical hardware requirements, you need to consider:

- ▶ The performance of the ETL process
- ▶ The performance and concurrency of the online reporting solution

While the ETL process is fixed regarding concurrency, the reporting needs will vary greatly. The demands on Primavera Data Warehouse and Primavera Analytics may change from day-to-day. For performance sizing of OBI, please refer to the technical documents for the specific component (BI Publisher, BI Server, and Answers/Dashboards). This document will focus on the performance of the ETL process and queries generated against the warehouse database (Star).

Monitoring Dimension and History Level Impacts on the Database

Primavera Data Warehouse includes slowly changing dimensions and daily level history for activity and resource assignment. If you choose to use slowly changing dimensions and these levels of history, they could cause the Primavera Data Warehouse database to grow quickly. For example, if you update 10 activities in a project and run the ETL process, 10 rows will be added in the activity slowly changing dimension table and in the assignment history and activity history fact tables.

You should plan the database and server configuration to ensure you can monitor database growth and allow for expansion based on the number of changes. If growth is exceeding what you expected in your environment, you can disable slowly changing dimensions if they are not required. See the *Primavera Analytics On-Premises Administration Guide* for more information.

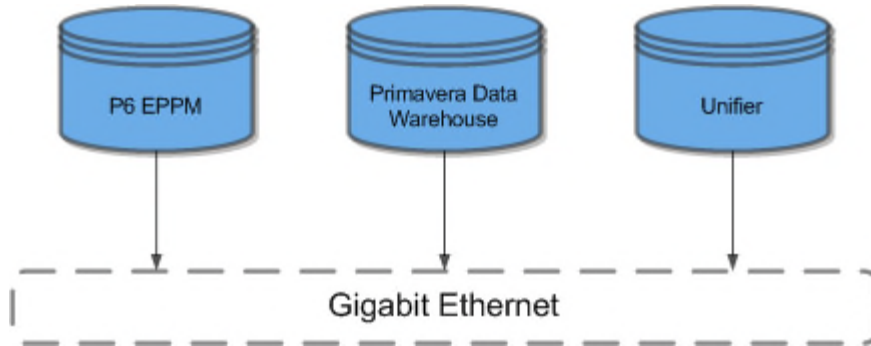
Consideration for ETL Performance

The ETL process for Primavera Data Warehouse was designed for multi-core processor systems. Instead of using a serialized process, Java is used to create a multi-threaded process to run concurrent threads of SQL*Plus and SQL*Loader. This can result in multiple run threads on the various servers. This also means that the process can be adversely affected by having to compete with other applications sharing the same resources. Therefore, an ideal configuration would have dedicated cores available for each component in the process and a dedicated database instance with sufficient SGA allocation and high throughput IO system.

This is an ideal configuration that is meant to minimize contention. By dedicating resources to each of the physical components, concurrent performance is maximized during peak usage. Different steps in the ETL process put a variety of load on each component.

Network

While there is an advantage to separating components, doing so assumes that the network connections between servers have high bandwidth and low latency. These servers should be on the same local area network with gigabit connections. Any increase in latency will have an effect on the ETL performance.



The Primavera Data Warehouse schema supports multiple P6 EPPM and Primavera Unifier data sources for a single data warehouse instance. With the internationalization of the corporate operations, it's likely that the data sources might be distributed across geographic regions and connected via corporate WAN instead of the high-speed, low-latency LAN. Data extraction is a high volume operation, and the bandwidth and network latency impact performance. Oracle has a 10-10 rule for the corporate WAN connections:

- ▶ Minimum network bandwidth: 10 mbps
- ▶ Maximum network latency: 10 ms

For a successful deployment, you should have network parameters that are better than these minimum requirements.

Memory

The large number of parallel processes running on large sets of data will significantly impact the demands on memory. You should run the components of the data warehouse system on 64-bit operating systems to allow for large memory allocations. Constraining memory quickly reduces performance.

The database servers need block buffer and individual process memory. These servers should always be setup using Dedicated Server (not Shared Server). For minimum required memory settings, see the Oracle Database documentation for your database version. Otherwise, let the database server manage its own memory.

The Java process on the ETL Process Server is running multiple threads in the same process. Only run with a 64-bit version of the JRE to allow for larger memory allocation. The maximum memory allocation for the Java process is configurable during setup (Max Heap Size). The default is 1 GB. This may be inadequate for many datasets and may cause failures in the ETLCalc process. Start with a minimum of 4 GB of memory for the Java process.

Extract and Load

Customized ETL processes are implemented for the data extraction and loading. These processes are completely PL/SQL based and use only resources on the database server. Multiple database threads can run PL/SQL anytime. These bulk PL/SQL inserts, updates, and deletes make the process resource intensive. Oracle recommends 10 or more cores on the database server, so each thread has an available CPU to run. The performance of each core will determine the speed (rows/second) of the extract and load processes.

Oracle Data Integrator (ODI)

For ODI daily ETL, there are some settings that may affect performance. The setting values are specific to the data source.

In the ETL_PARAMETER table:

```
select p_1 from ETL_PARAMETER where p_feature in
('odi.spread.percent.threshold', 'odi.spread.update.percent') and
datasource_id=9;
```

The default value is 25. For Medium and Large databases, reduce this number. For example, set to value 10 for medium database and 3 for large database. If an entry is not there, insert one for each datasource as follows:

```
insert into ETL_PARAMETER (p_feature, p_1, datasource_id) values
('odi.spread.percent.threshold', 3, 9);

insert into ETL_PARAMETER (p_feature, p_1, datasource_id) values
('odi.spread.update.percent', 3, 9);
```

Summary of Physical Hardware Sizing

Review the Oracle Fusion Middleware System Requirements and Specification at http://docs.oracle.com/cd/E23104_01/sysreqs1221/sysrs.htm#CHDGAAEJ. Then use the guidelines in this section to plan the physical hardware for Primavera Data Warehouse.

Size of Database

The size of the P6 EPPM or Primavera Unifier database will affect performance and the performance of the ETL process, since all records must be processed. The size of the database can also have some impact on the ETL process, since more project data will likely translate into more usage and more records in the database.

Amount of Change

You should monitor the performance of the Px services. While the database size plays a role, the volume of changes made will affect performance most. Even a small user community can generate a large amount of changes with functions like copy/paste, delete project, and create baseline, as well as simple changes to global resources like calendars. You should monitor the amount of changes made prior to installing the products to better plan the implementation.

Primavera Data Warehouse always runs a full ETL process. Fresh data is pulled from the P6 EPPM schema and existing history table data are preserved during the ETL process. You must decide whether to implement a history data purge process.

Complexity of Project Data

If you are using P6 EPPM as a data source, consider the data in the database. A very complex EPS/WBS structure will increase the processing time because these hierarchical elements require global processing even when the changes are small. The average size of activities can increase the memory requirements of the calculation process. Larger projects will cause more volume of change because the entire project is recalculated based on any changes within the project. This does not include changes to non-scheduling data, such as activity codes and UDFs.

Expectations and Operations

You may expect the ETL to complete in a smaller time frame for operational purposes. This expectation could increase the hardware requirements. The following sections detail requirements for each database size.

Small/Medium Database Deployment

ETL Server and Primavera Data Warehouse database on same computer

CPU	Quad Core 3.46 GHz or equivalent
RAM	8 GB

Notes:

- These requirements assume light Business Analytics (Primavera Data Warehouse database) usage. If that is not the case, use the Large Database Deployment model and allocate a separate server for the Primavera Data Warehouse database.
- You should add 4 GB of memory if you installed OBIEE on the same computer.

Large Database Deployment

ETL Server

CPU	Quad Core 3.46 GHz or equivalent
RAM	8 GB

Primavera Data Warehouse Database

CPU	8 Core 3.46 GHz or equivalent
-----	-------------------------------

RAM	16 GB
-----	-------

Database Servers

	Primavera Data Warehouse Database
TableSpace Autoextent	Yes
Recommended Temp Tablespace	Minimum 2 files with file size set at OS max
Recommended UNDO Tablespace	Minimum 2 files with file size set at OS max
Minimum Space	Same as P6 EPPM database
SGA	8 GB

Planning Revisited

Timing Estimation

The following timing numbers are only an approximation for P6 EPPM. The actual numbers for your deployment depend on the data structure and the hardware used.

Note: For planning and sizing purposes, Primavera Unifier databases are considered "small".

Initial ETL

	Number of Objects		
	Small	Medium	Large
Projects	200	1,000	15,000
Activities	100,000	500,000	4,000,000
Resources	5,000	5,000	5,000
Resource Assignments	30,000	100,000	2,000,000

Codes Mapped	1	1	200
UDFs Mapped	0	0	200
Project with History (percentage of total)*	5.00%	5.00%	50.00%
Backlog of ETL runs	--	--	--
STAR_DATx cumulative size**	3.5 GB	8 GB	90 GB
STAR_HSTx cumulative size**	< 1 GB	< 1 GB	< 1 GB
Full ETL Approximate completion time	< 1 hr	< 1 hr	~ 7 hrs
LOAD_PLAN_INITIAL_LOAD	< 1hr	< 1 hr	5 hrs

Incremental ETL - 2 months of history and Slowly Changing Dimensions

	Number of Objects		
	Small	Medium	Large
Projects	200	1,000	15,000
Activities	100,000	500,000	4,000,000
Resources	5,000	5,000	5,000
Resource Assignments	30,000	100,000	2,000,000
Codes Mapped	1	1	200

UDFs Mapped	0	0	200
Project with History (percentage of total)*	5.00%	5.00%	50.00%
Backlog of ETL runs	60 days	60 days	60 days
STAR_DATx cumulative size	8 GB	17 GB	255 GB
STAR_HSTx cumulative size	< 1 GB	< 1 GB	60 GB
Full ETL Approximate completion time	~ 2 hrs	~ 3 hrs	~ 11 hours
LOAD_PLAN_DAILY	< 1 hr	< 2 hrs	< 4 hrs

*In performance runs, history level is all Activity Levels. This is a worst case scenario.

**Oracle data files: depending on how you manage your tablespace, you can have a different number of data files.

Factors Affecting ETL Runtime

All the following can affect ETL runtime.

Projects Published The number of projects and associated project data will have a direct correlation to the duration of the ETL run time. Carefully consider which projects require loading to the Primavera Data Warehouse database by filtering for these projects. Some considerations may be active projects or projects that meet some specific criteria by using a project filter.

Spread Interval In P6 EPPM, when configuring the service settings, the administrator must define the spread interval for each project. It is defined as a starting date to the current date plus a rolling interval. The definition of this interval has a direct correlation to the amount of spread data in Star: the larger the interval, the greater the number of spread records for each project. Oracle recommends keeping the rolling interval to a couple years from the current data.

Traditional History This feature keeps history at a level defined within a project at a specific interval. The number of projects that meet the designated history level and the intervals that have been crossed throughout the life cycle of the Primavera Data Warehouse database will determine the size of these history tables. You should consider how features can optimize performance, such as:

- ▶ Selectively determining the number of projects that require history.
- ▶ Determining the level of storage granularity for each project with history. The higher the granularity, the fewer rows there will be over time.
- ▶ Turning off project history over time as projects end or close to minimize the cost of writing records for projects that are not active anymore.
- ▶ Noting requirements during the initial configuration to correctly size the number of partitions for the Primavera Data Warehouse database and the months to include in each partition. An even distribution of data per partition is optimal to increase performance.

Activity Level History & Slowly Changing Data In Primavera Data Warehouse, Activity Level History has been coupled with features that can impact performance of the ETL process. The most important of these features is Slowly Changing Data. When a project has Activity Level History enabled, changes are tracked at the field level and new dimension and fact records are created with each change. You should limit the projects with this history setting to ones that need this level of granularity and require tracking of changed data over time. The more projects with this setting, the more records will be stored in the activity history fact table and the slowly changing dimensions and fact tables.

Burn Down and Work Planning These two features are driven by the project having Activity Level History and a set of predefined user-defined fields with appropriate values. The calculation of these projects with this feature enabled can be costly, so you should minimize the number of projects with these features.

The Number of User-Defined Fields and Codes mapped During the initial configuration, select user-defined fields and codes to include within the Primavera Data Warehouse database. Note the requirements to determine which user-defined fields and codes are needed. The more UDFs and codes that are mapped, the more they will affect the STARETL process.

Steps in the STARETL process that have the most significant runtime due to settings:

ETL Step	Definition	Affecting Factors	Considerations
ActivityResourceAssignmentSpreads	Load activity and resource assignment spreads (fact_load)	<ul style="list-style-type: none"> ▶ number of projects published ▶ spread interval defined ▶ number of projects with activity level history ▶ changes to project data with activity level history over time 	<ul style="list-style-type: none"> ▶ using project filter ▶ reducing spread interval ▶ evaluating projects that need activity level history to either turn off or move to different history setting

MergeProjectHistory	Merge Project History (hist_merge)	<ul style="list-style-type: none"> ▶ number of projects with history defined ▶ size of existing history ▶ history intervals 	<ul style="list-style-type: none"> ▶ limiting projects with history ▶ increasing interval of history
SourceExtract	Extract data from source database (source_extract)	<ul style="list-style-type: none"> ▶ amount of data published 	<ul style="list-style-type: none"> ▶ using project filter ▶ thread count increase
DimensionLoad	Load Dimension Tables (dim_load)	<ul style="list-style-type: none"> ▶ amount of data published ▶ projects with activity level history 	<ul style="list-style-type: none"> ▶ using project filter ▶ re-evaluation project with activity level history

Planning for the Impact of P6 EPPM on Primavera Data Warehouse

You have many different configuration options you can deploy to accommodate your environment. The configuration that works best will depend on the data size, data requirements, environment, and servers available.

The data flow from P6 can affect your Primavera Data Warehouse.

Choosing Whether to Use Partitioning

The following table shows supported features with and without partitioning.

Feature	Enterprise Edition Without Partitioning	Enterprise Edition With Partitioning
Multiple Data Sources		x
Activity Level History		x
WBS Level History	x	x
Assignment Level History		x
Project Level History	x	x

Burndown Subject Area		x
Work Planning Subject Area		x
Small Database Size	x	x
Medium Database Size		x
Large Database Size		x
Row Level Security	x	x

Effect of the P6 Extended Schema on Primavera Data Warehouse

The P6 Extended Schema is the primary source of data for Primavera Data Warehouse. You can schedule services to continuously update data to allow for near-real time reporting, which allows the database to cover more data. The amount of data populated in the P6 Extended Schema will have a cascading affect on the size of the Primavera Data Warehouse database and the duration of the ETL process.

Date Range

In P6, you can define the date range for the Extended Schema. The range begins with the implementation date and a determined date range (for example, 3 years). Extending the range too far into the future will create large spread ranges. If you use the default range of 3 years, P6 will create a rolling window. As the services run, the window will continue to move. For example, January 1, 2012 and January 1, 2015 will become January 2, 2012 and January 2, 2015. If you have projects that last for three years, the projects will naturally fall in line and give you the spread data for this time frame.

Spread data represents a row in the database for each activity that has work on that day and for every activity in every project that is published. For example, if you have a project that has data reaching out 10 years and has 100 activities, you will have a spread row for 100 activities x 365 days per year x 10 years. Using this formula, data can add up quickly. The Extended Schema handles this and the ETL process quickly moves this data.

If you do not need these projects in Primavera Data Warehouse for analytical reporting, you can still use them in the Extended Schema for reporting. You can use a filter to decide which projects to include in the ETL process. By default, all projects with the **Enable Publication** setting in the Extended Schema will transfer to Primavera Data Warehouse.

Enable Publication

If you want to enable publication for projects, you must select the **Enable Publication** setting in P6. When you enable publication the Project Service will retrieve and calculate this project. The project will then be part of the Extended Schema. The Project Service will monitor how many changes have been made on the project and how long since the project was published.

Each project service run will write the number of changes on that project to a table. If it crosses this threshold, the project will queue for processing. If this project has any changes, it will be processed. Primavera Data Warehouse will use the Enable Publication setting to determine which projects are pulled into the Primavera Data Warehouse database.

History Settings on Project Level

If you are using P6 EPPM 8.2 SP1 or later, the project history level settings are available in P6 and can be set project by project; history will populate up from activity to WBS to project. If you set history at the Activity level, P6 will also populate WBS and Project history. You can set history for month, financial period, quarter, or year. Selecting Activity-level history means you can capture daily activity and resource assignment data regardless of the periodic setting in P6.

Define history levels selectively. History will impact how much data and what level the history fact tables in Primavera Data Warehouse will process. Depending on your database size, enabling Activity level history for all projects could cause a long ETL process time. Turn on history only for those projects that require history. When projects end, turn off history settings to prevent performance issues.

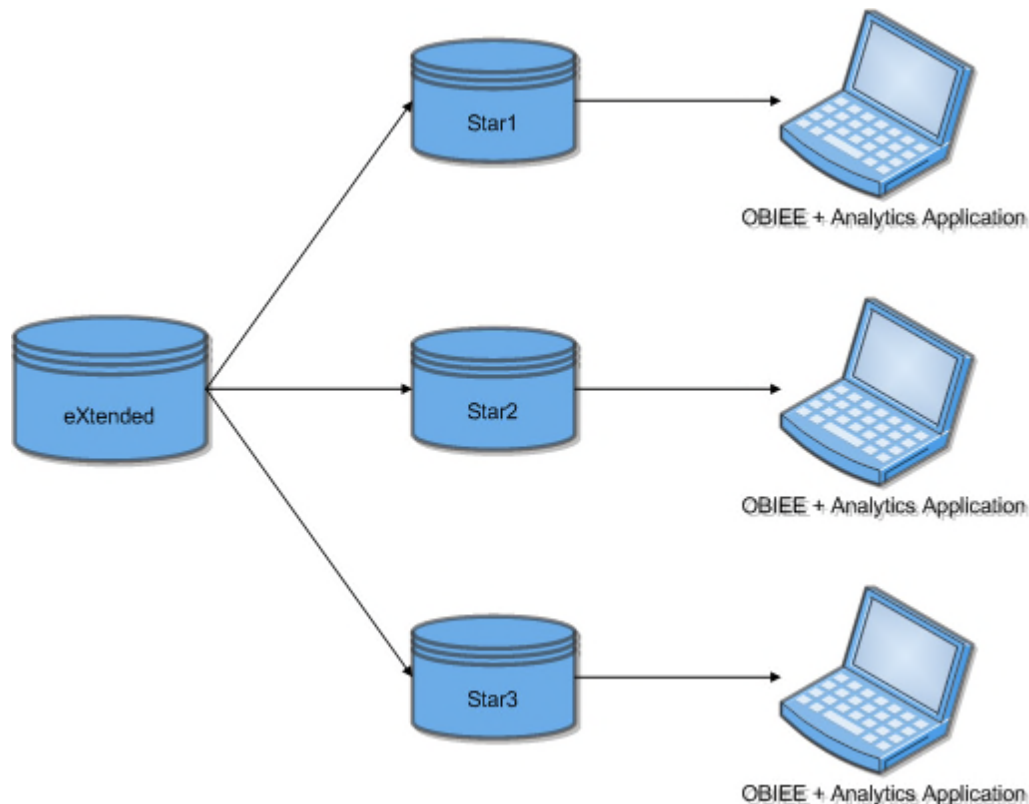
History is captured and rolled over in a merge to avoid overwriting existing history. This capture and merge can cause the ETL process to slow slightly after several historic period captures until crossing over a new partition. Partitioning is available to help performance and move older history into a new partition and help decrease ETL run time. Partition interval ranges are predefined in tables and index scripts in the Primavera Data Warehouse installation\scripts directory. See the *Primavera Analytics On-Premises Administration Guide* for more information.

Slowly Changing Dimensions

Projects that have HIST_LEVEL HL_Task will be part of Slowly Changing Dimensions.

Multiple Primavera Data Warehouse Environments

If you require low granularity (Activity, Daily) for a large number of projects, you may want to use multiple Primavera Data Warehouses. Multiple Primavera Data Warehouses can help decrease the run time and separate data into different STARETL processes. To separate projects, you need to add a defining characteristic (portfolio, project code assignment, and so on) to the project in P6. The following diagram shows how the different Primavera Data Warehouse environments will connect to their own OBI environment.



You can filter data for Primavera Data Warehouse. The filter sets a Select statement that retrieves the project IDs for only the projects you want in a single Primavera Data Warehouse. This can be any criteria, such as portfolio or project code assignment. This filter mechanism overrides the default process of gathering all the project IDs for each project in the Extended Schema that has Enable Publication set. See the *Primavera Analytics On-Premises Administration Guide* for details on defining filters.

Creating Multiple Data Sources for a Single Primavera Data Warehouse Database

You can combine more than one Extended Schema into a single Primavera Data Warehouse database. This enables you to have multiple P6 installations and view the combined data in one Primavera Data Warehouse. You can also set up filters that enable you to view data from just one data source.

Consider the following when using multiple data sources:

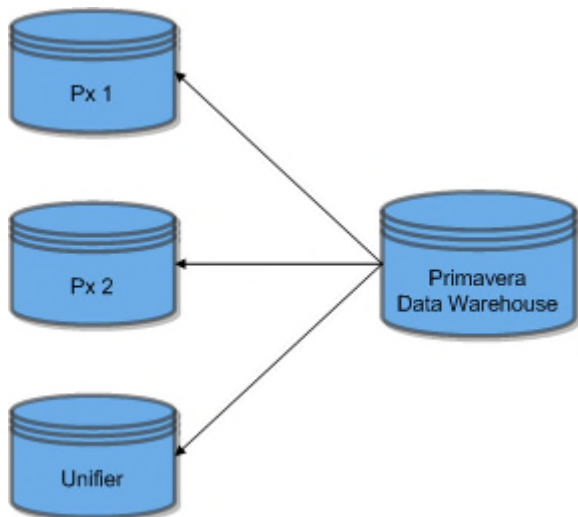
- ▶ Ensure that the names of resources, projects, or EPSs match across data sources.
- ▶ Store the data in the same dimensions and facts for each data source.
- ▶ Insert rows with unique row IDs for each data source.

Note: You cannot run the ETLs for the data sources at the same time. Each additional data source requires a unique installation with its own configuration information. Each installation will have its own staretl. The staretl processes cannot run at the same time because the unified Primavera Data Warehouse will still use the same internal staging tables. If multiple ETL processes were running at the same time, they could overwrite each other in the staging tables and cause failures or data corruption.

Setting up Multiple Data Sources

You can create up to 99 data sources. You would create each data source individually and connect the specific Extended Schema where it should pull data. If you go above three data sources, you must adjust the scripts to accommodate the extra data sources.

You must determine how many data sources you require before you begin running ETL processes. You can also apply a filter to each data source to pull certain data from the extended schema. You must then schedule the ETL processes to run so they don't impact each other. The resulting environment could look similar to the diagram below:



For details setting up multiple data sources, see the *Primavera Data Warehouse Installation and Configuration Guide*.

Filtering by Data Source in Analytics

When creating analyses in OBI, a filter can be applied where:

data source id = (data source number)

In OBI, you can find a data source ID field under Project – General in most subject areas. You can apply the data source ID to any Analysis and set a value to limit the output to only that desired data source. You will see a similar data source ID in resource-related Subject areas under Resource – General.

Single Primavera Data Warehouse, Single Data Source, Multiple ETLs

In this case, you have one primary data source pulling from the Extended Schema (Px). There is only one Primavera Data Warehouse, but you will have different ETL processes. You may need different ETL processes to update certain projects for a different company division at a different time. Because the ETL process is typically meant to run once a day due to limitations of time and staging tables, you could use filtering and multiple data sources inside of one Primavera Data Warehouse instance.

Caution: ETL processes cannot run at the same time. You cannot overlap ETL processes because the staging tables inside Primavera Data Warehouse are shared and any overlap can cause data corruption.

Appendix A: Checklists for Primavera Analytics and Primavera Data Warehouse

Review the details of this checklist before installing Primavera Analytics. Use the checklist to ensure you successfully install Primavera Analytics. The checklist contains details on environment settings you need to consider, such as memory and disk space.

Pre-installation for an Oracle Platform

Ensure you:

Check	Item
	Downloaded Primavera Analytics and Primavera Data Warehouse software for the correct operating system. (See the Oracle Software Delivery Cloud.)
	Installed the correct JRE version (see the <i>Tested Configurations</i> document).
	Installed the correct JRE platform (x86 versus x64).
	Can connect to the P6 EPPM and/or Primavera Unifier database from this computer.
	Installed P6 EPPM and/or Primavera Unifier and verified which release you have. Note: See the <i>Tested Configurations</i> for supported versions.
	Have the same Oracle database version on all servers.

	<p>Enabled partitioning on the Primavera Data Warehouse instance.</p> <p>Note: Oracle recommends you use partitioning if the P6 EPPM and/or Primavera Unifier database is larger than a small-sized database as defined in <i>Planning for the Impact of P6 EPPM on Primavera Data Warehouse</i> (on page 31) or if multiple data sources are connected to the Primavera Data Warehouse.</p>
--	---

Pre-installation for a Microsoft SQL Server Platform

Ensure you:

Check	Item
	Downloaded Primavera Analytics and Primavera Data Warehouse software for the correct operating system. (See the Oracle Software Delivery Cloud.)
	Installed the correct JRE version (see the <i>Tested Configurations</i> document).
	Installed the correct JRE platform (x86 versus x64).
	Can connect to the P6 EPPM database from this computer.
	Installed P6 EPPM and verified which release you have. Note: See the <i>Tested Configurations</i> for supported versions.
	Installed the Oracle Gateway.
	Can connect the Oracle Gateway to the P6 EPPM database. Note: To determine whether you have a connection, run a SELECT statement through Oracle Gateway.

Pre-installation for Configuring the ETL Server

Ensure you:

Note: This section is based on large implementations. Refer to the *Summary of Physical Hardware Sizing* (on page 25) section for details on small and medium implementations.

Check	Item
	Reviewed the <i>Primavera Analytics Planning and Sizing Guide</i> for sizing and spacing guidelines.

	Can connect to the P6 EPPM and/or Primavera Unifier database. Note: If you are using Microsoft SQL Server, ensure you can connect to the P6 EPPM database through the Oracle Gateway.
	Can connect to the Primavera Data Warehouse database instance.
	Configured TNSNAMES correctly with all instances.
	Defined the ORACLE_HOME path on the server running the ETL scripts so it can find SQLLDR.
	Can connect JDBC and OCI on the ETL server. Note: You can test the connection with a sample Java application.
	Verified the user running scripts on the server has Read and Write permissions to install the directory. Note: On UNIX systems, create and install the database as 'Oracle' user.
	Placed all servers in same data center with a Gigabit Ethernet connection between servers.
	Verified each ETL Process server has 8-12 core processors.
	Verified each database server has 8-10 core processors.
	Are running the ETL Server on 64-bit machine to allow for large memory allocations.
	Set the Java Max Heap setting to at least 4 GB of memory. Allow for more memory, if available, for the Java process.
	Set the Bulk Load files location from the ETL process to a location with a large amount of available disk space. Note: These files can become large depending on the size of the database. Available space should range from a few gigs to many tens of gigabytes.
	Are using a supported Operating System. See the <i>Tested Configurations</i> document for supported operating systems.

Pre-installation for Oracle Data Integrator (ODI)

Ensure you:

Check	Item
	Downloaded <i>Primavera Analytics and Primavera Data Warehouse</i> software for the correct operating system. (See the Oracle Software Delivery Cloud.)

	Installed the correct JRE version (see the Tested Configurations document).
	Installed the correct JRE platform (x86 versus x64).
	Can connect to the P6 EPPM and/or Primavera Unifier database from this computer.
	Install Oracle Data Integrator and setup ODI Agent.
	Installed P6 EPPM and/or Primavera Unifier and verified which release you have. (see the Tested Configurations document).
	Have the same Oracle database version on all servers.
	Enabled partitioning on the Primavera Data Warehouse instance. Note: Oracle recommends you use partitioning if the P6 EPPM and/or Primavera Unifier database is larger than a small-sized database as defined in <i>Planning for the Impact of P6 EPPM on Primavera Data Warehouse</i> (on page 31) or if multiple data sources are connected to the Primavera Data Warehouse.

Pre-installation for the Primavera Data Warehouse Database

Ensure you:

Check	Item
	Reviewed the <i>Primavera Analytics Planning and Sizing Guide</i> for sizing and spacing guidelines.
	Have a tablespace called STAR_DAT1.
	Have a tablespace called STAR_HST1.
	Set the tablespace with Auto Extend On.
	Set the tablespace extent management to Local.
	Set the character set to UTF8 or WE8MSWIN1252. Primavera Analytics also supports the AL32UTF8 character set.
	Set the character set to the same character set used in the P6 EPPM and/or Primavera Unifier database.
	Set the Primavera Data Warehouse tablespace to at least the same size as the P6 EPPM and/or Primavera Unifier database.
	Run the queries to validate database settings. See <i>Appendix B: Database Scripts For Validations</i> (on page 43) for details.
	Set datafiles to autoextensible.
	Set maxbytes to the OS's maximum, which is typically 32 GB with minimum 2 files. Use 3 to 5 files for larger systems.

	Run the queries to validate the UNDO Tablespace settings. See Appendix B: Database Scripts For Validations (on page 43) for details.
	Run the queries to validate the SGA settings on the database. See Appendix B: Database Scripts For Validations (on page 43) for details. Note: Larger systems require 8 GB.
	Run the queries to validate the Archive Log Mode on the database. See Appendix B: Database Scripts For Validations (on page 43) for details. Note: This should be in NON archive log mode.
	Run the following SQL to verify the tablespace: <pre>select s.tablespace_name, f.file_name, f.maxbytes, f.autoextensible from dba_data_files f inner join dba_tablespaces s on s.tablespace_name = f.tablespace_name where s.tablespace_name like '%STAR%';</pre>
	Have both block buffer and individual process memory available.
	Setup the database instance using Dedicated Server (not Shared Server).
	For the MEMORY_TARGET variable, let the database server manage its own memory.
	Can connect JDBC and OCI on the Star server. Note: You can test the connection with a sample Java application.
	Decided whether there will be multiple data sources. The sizing requirements will apply for each individual data source included.
	Verified that you are using one of the following database versions. See the <i>Tested Configurations</i> document for information on supported databases.

Installation: Installer And Configuration

Ensure you:

Check	Item
	Reviewed the <i>Primavera Data Warehouse Installation and Configuration Guide</i> and the <i>Primavera Analytics Installation and Configuration Guide</i> .
	Set the JAVA_HOME and PATH definition. Note: The config.cmd or config.sh needs JAVA_HOME set in order to launch. For example, JAVA_HOME=C:\Program Files\Java\jre1.8.0_72 (with Windows) or PATH=\$PATH:/u01/jre1.8.73/bin (with UNIX or Linux).

	<p>Determine your date range for your Primavera Data Warehouse.</p> <p>Note: Determine your date range before running the installer. This date range will include spread data and the start date and rolling date range of your data warehouse. See the <i>Primavera Analytics On-Premises Administration Guide</i> for details.</p>
	<p>Determine a reasonable date range setting.</p> <p>Note: If you set a large data range this will result in a very large amount of spread data. A large amount of spread data requires more database space, more hard drive space for temp files, and increased run times. Consider your date range carefully. Spreads outside of the date range will combine into one bucket at the beginning and end of the date range.</p>
	<p>Determine your Project Trend Interval.</p> <p>Note: Project Trend Interval will determine how to store your project history. See the <i>Primavera Analytics On-Premises Administration Guide</i> for details.</p>
	<p>Determine which activity, project, and resource codes you will use before installation.</p> <p>Note: Dynamic codes are available for slicing data in OBI. Code hierarchies are also supported if you select them when you install this application.</p> <ul style="list-style-type: none"> ▶ Matching Criteria/Regular expression: The name of the code as registered in Project Management. ▶ Name: The display name for the code name label in OBI. ▶ Description: The display name for the code value label in OBI.
	<p>Determine which activity, project, and resource UDFS you will use before installation.</p> <ul style="list-style-type: none"> ▶ Matching Criteria/Regular expression: The name of the UDF as registered in P6 and/or Primavera Unifier. ▶ Name: The display name for the UDF name label in OBI. ▶ Description: The display name for the UDF value label in OBI.
	<p>Consider the available disk space when you set the Logging level.</p> <p>Note: If DEBUG is set, the logs can become large. Oracle recommends INFO for a logging level unless troubleshooting is an issue.</p>
	<p>Set the Java Max Heap setting to at least 4 GB of memory. Allow for more memory, if available, for the Java process.</p> <p>Note: The default for the Maximum Java Heap Size (MB) parameter is 1,204MB. The minimum value is 512MB. The value assigned must be a multiple of 512MB. This guide references 1 GB, but notes that 1 GB may be inadequate for many datasets and may cause failures in the ETLCalc process. Oracle recommends starting with a minimum of 4 GB of memory for the Java process.</p>
	<p>Have the P6 EPPM and/or Primavera Unifier privuser username/password information available.</p>

	Have the Primavera Data Warehouse instance and system username/password information available.
	Determine which P6 EPPM and/or Primavera Unifier users will be reporting users. Note: Reporting users will have a database user created for them. That means you must set the report_user_flag field to 'Y' on the USERS table in the P6 EPPM and/or Primavera Unifier database for all those users who will become reporting users.
	Define Financial Periods before running Primavera Data Warehouse processes. Note: If you will set the Project Trend intervals based on the Financial Period, you must set the Financial Period before running Primavera Data Warehouse processes.
	Set the Bulk Load files location for the ETL process to a location with a large amount of available disk space. Note: These files can become large depending on the size of the database. You need ranges from a few gigs to many tens of gigabytes of available space.
	Set Oracle database user requirements for users and passwords when creating the Oracle stageuser and staruser.

Installation: Running The Processes

Ensure you:

Check	Item
	Grant EXECUTE to PUBLIC for the standard Oracle SYS.DBMS_RANDOM package to run the RunETL Process.
	Locate the installation home directory that contains the staretl.bat or staretl.sh .
	View the log and html file for the ETL process in the installation home directory to view the status of the process and verify it completed successfully.

Post-installation: After The Processes Complete

Ensure you:

Check	Item
	Run staretl.bat or staretl.sh .

	In <PDW Install Folder>\starletl_homes\staretl<id>\log , review the most recent Status Log and ETL Log to ensure processes completed successfully.
	Ensure the system created a staretl.bat or staretl.sh and the appropriate log files. See the <i>Primavera Data Warehouse Installation and Configuration Guide</i> for more information on logs.

Appendix B: Database Scripts For Validations

Use the following scripts to validate your database:

Script	Description
SGA (Memory) Validation	Requires 8 GB for larger systems. Script: SQL> SELECT * FROM v\$sgainfo; SQL> show sga
Check archive log mode	For ERDB, the database should be in non-archive mode. Script: SQL> SELECT log_mode FROM v\$database;
Processes Init Parameter	Requires a minimum of 300. Script: SQL> show parameter process
Temp Tablespace Validation	Requires a minimum of two files with file size set at the OS maximum, typically 32 GB. Larger systems should have three files for the temp tablespace. Script: SQL> select TABLESPACE_NAME, FILE_ID, BYTES_USED, BYTES_FREE from V\$TEMP_SPACE_HEADER;

<p>UNDO Tablespace Validation</p>	<p>Set the datafile to be autoextensible. Set the maxbytes to the OS maximum, typically 32 GB. Requires a minimum of two files; three to five files for larger systems. Script:</p> <pre>SQL> select s.tablespace_name, f.file_name, f.maxbytes, f.autoextensible from dba_data_files f inner join dba_tablespaces s on s.tablespace_name = f.tablespace_name where s.tablespace_name like '%UNDO%';</pre>
<p>Tablespaces for Star</p>	<p>Set the datafile to be autoextensible. Set the maxbytes to the OS maximum, typically 32 GB. Requires a minimum of two files; three to five files for larger systems. Script:</p> <pre>SQL> select s.tablespace_name, f.file_name, f.maxbytes, f.autoextensible from dba_data_files f inner join dba_tablespaces s on s.tablespace_name = f.tablespace_name where s.tablespace_name like '%STAR%';</pre>

Copyright

Oracle Primavera Analytics Planning and Sizing Guide for On-Premises

Copyright © 1999, 2020, Oracle and/or its affiliates.

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

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

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.