ORACLE
PEOPLESOFT ENTERPRISE

PeopleSoft Enterprise EPM 9.1 Enterprise Performance Management Fundamentals PeopleBook

April 2010

ORACLE

PeopleSoft Enterprise EPM 9.1 Enterprise Performance Management Fundamentals PeopleBook
SKU epm91penw-b0410

Trademark Notice

License Restrictions Warranty/Consequential Damages Disclaimer

Warranty Disclaimer

Restricted Rights Notice

Hazardous Applications Notice

Third Party Content, Products, and Services Disclaimer

Contents

Chapter 6

Part 3

Bringing Source Data Into EPM Using Extract, Transform, and Load (ETL)

Chapter 8

Chapter 9

Part 4

Setting Up the Operational Warehouse - Enriched for EPM Analytical Applications

Chapter 16

Chapter 17

Chapter 18

Oracle's PeopleSoft Enterprise Performance Management Fundamentals 9.1 Preface

Welcome to the *PeopleSoft Enterprise Performance Management Fundamentals 9.1 PeopleBook*, which describes how to set up Enterprise Performance Management (EPM) and use the EPM Foundation toolset.

This preface discusses:

- PeopleSoft products.
- Common elements used in the EPM documentation set.
- Deferred processing in EPM.

PeopleSoft Enterprise Products

This PeopleBook refers to these PeopleSoft products:

- Oracle's PeopleSoft Enterprise Activity-Based Management
- Oracle's PeopleSoft Enterprise Campus Solutions Warehouse
- Oracle's PeopleSoft Enterprise Customer Relationship Management Warehouse
- Oracle's PeopleSoft Enterprise Financial Management Solutions Warehouse
- Oracle's PeopleSoft Enterprise Global Consolidations
- Oracle's PeopleSoft Enterprise Human Capital Management Warehouse
- Oracle's PeopleSoft Enterprise Planning and Budgeting
- Oracle's PeopleSoft Enterprise Scorecard
- Oracle's PeopleSoft Enterprise Supply Chain Management Warehouse

Common Elements Used in the PeopleSoft EPM Documentation Set

This section lists common elements used in PeopleSoft EPM.

SetID Provides the ID code for a tableset. A tableset is a group of tables (records) necessary to define your company's structure and processing options.

Effective Date	Establishes the date the row in the table becomes effective. It determines when you can view and change the information. Pages and batch processes that use the information use the current row.
Status	Indicates whether a row in a table is active or inactive. You cannot select inactive rows on pages or use them for running batch processes
Description	Enables you to input freeflow text, up to 30 characters, that describes what you are defining.
Run Control ID	Identifies specific run control settings for a process or report.
Report ID	Identifies the report.
Program Name	Provides the Enterprise Performance Management program name for which you are running the report or process.
When	Specifies the frequency with which you want to run a process. You can choose *Once*, *Always*, or *Don't.*
Last Run On	Indicates the date the report or process was last run.
As Of Date	Indicates the last date for which the report or process includes data.
Scenario ID	Provides an identifier for a specific scenario.
Model ID	Provides an identifier for a model. A model uniquely identifies the types of data you want to include in a scenario. For example, you might want to review revenue by region—a very high-level scope. Or, if you use Activity-Based Management, you might want to review only those activities that relate to a certain product line for certain types of resources—a very narrow scope.
Fiscal Year	Specifies the fiscal year for your scenario or process run.
Period	Specifies the accounting period for the object being defined or process being run.
Job ID	Specifies an instance of an engine.

Deferred Processing in PeopleSoft EPM

Several pages in PeopleSoft EPM operate in deferred processing mode. Most fields on these pages are not updated or validated until you save the page or refresh it by clicking a button, link, or tab. This delayed processing has various implications for the field values on the page. For example, if a field contains a default value, any value that you enter before the system updates the page overrides the default. Another implication is that the system updates quantity balances or totals only when you save or otherwise refresh the page.

PeopleBooks and the Online PeopleSoft Library

A companion PeopleBook called PeopleBooks and the Online PeopleSoft Library contains general information, including:

- Understanding the PeopleSoft online library and related documentation.
- How to send PeopleSoft documentation comments and suggestions to Oracle.
- How to access hosted PeopleBooks, downloadable HTML PeopleBooks, and downloadable PDF PeopleBooks as well as documentation updates.
- Understanding PeopleBook structure.
- Typographical conventions and visual cues used in PeopleBooks.
- ISO country codes and currency codes.
- PeopleBooks that are common across multiple applications.
- Common elements used in PeopleBooks.
- Navigating the PeopleBooks interface and searching the PeopleSoft online library.
- Displaying and printing screen shots and graphics in PeopleBooks.
- How to manage the PeopleSoft online library including full-text searching and configuring a reverse proxy server.
- Understanding documentation integration and how to integrate customized documentation into the library.
- Glossary of useful PeopleSoft terms that are used in PeopleBooks.

You can find this companion PeopleBook in your PeopleSoft online library.

Part 1

Getting Started with Enterprise Performance Management

Chapter 1

Getting Started With PeopleSoft Enterprise Performance Management

This chapter provides an overview of PeopleSoft Enterprise Performance Management (EPM) and discusses:

- EPM Integrations.
- EPM Implementation.

PeopleSoft EPM Overview

EPM is a packaged data warehousing platform that serves both as a repository of enterprise information for reporting and analysis and as the foundation for the PeopleSoft EPM Warehouses and Analytical Applications. EPM provides the tools necessary to gather data from transactional, legacy, and external data sources, stage, store, and enrich that data, and make the information available for analysis.

See Also

Chapter 2, "Understanding PeopleSoft Enterprise Performance Management," page 5

PeopleSoft EPM Integrations

PeopleSoft EPM is the central repository for the PeopleSoft EPM Warehouses and Analytical Applications, and fully integrates with:

- PeopleSoft Enterprise transactional systems
- IBM WebSphere DataStage
- EPM Warehouses
- PeopleSoft EPM Analytical Applications

PeopleSoft EPM Implementation

PeopleSoft Setup Manager enables you to generate a list of setup tasks for your organization based on the features that you are implementing. The setup tasks include the components that you must set up, listed in the order in which you must enter data into the component tables, and links to the corresponding PeopleBook documentation.

Other Sources of Information

In the planning phase of your implementation, take advantage of all PeopleSoft sources of information, including the installation guides, table-loading sequences, data models, and business process maps. A complete list of these resources appears in *PeopleSoft and the Online Library,* with information about where to find the most current version of each.

See Also

Chapter 3, "Implementing PeopleSoft Enterprise Performance Management," page 31

Enterprise PeopleTools PeopleBook:PeopleSoft Setup Manager

Chapter 2

Understanding PeopleSoft Enterprise Performance Management

This chapter provides an overview of Enterprise Performance Management (EPM) and discusses:

- EPM Architecture.
- Extract, Transform, and Load (ETL) in EPM .
- Operational Warehouse - Staging (OWS).
- Operational Warehouse - Enriched (OWE).
- Multidimensional Warehouse (MDW).
- EPM Foundation Toolset.
- PeopleSoft EPM Analytical Applications.
- PeopleSoft EPM Warehouses and Reporting.

Overview

PeopleSoft EPM is a comprehensive, integrated analytic business solution designed to increase the efficiency of your organization. PeopleSoft EPM helps your organization achieve operational excellence by providing insight into the information you need to drive predictability, accountability, and manage operational risk. EPM enables you to produce detailed activity analyses and resource plans, understand the cause-and-effect relationship between cost and behavior, organize strategic thinking and performance measurement, use continuous, collaborative forecasting to manage the plan and budget in real-time, and clearly communicate strategy and success measures.

EPM is supported by data warehouses, related data models, robust infrastructure and metadata, and the EPM Foundation toolset. EPM provides all the necessary tools to gather and manage data from PeopleSoft, legacy, and external data sources, enrich that data, and store it in an intuitive analytic context for you to analyze in a variety of ways and at a variety of levels. EPM enables you to deliver a single, accurate view of information across your organization.

PeopleSoft EPM Architecture

PeopleSoft EPM Warehouses and Analytical Applications are built on a foundation of specialized data warehouses, target warehouse tables, ETL jobs, metadata, and other prepackaged content that enable complex analysis and reporting of your data.

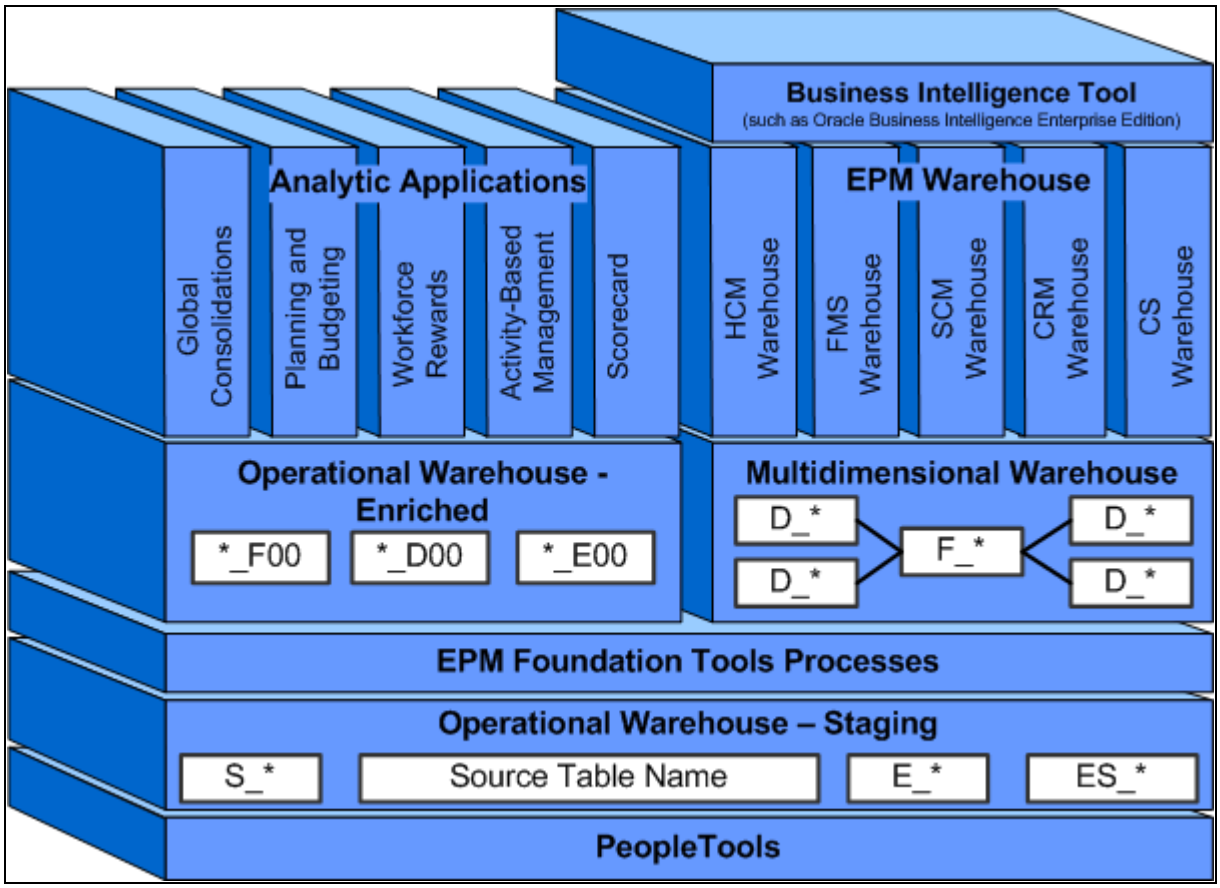
EPM target warehouse tables provide a way to consolidate and store your source transaction data. EPM target warehouse tables reside in two high-level data warehouse structures:

- the *Operational Warehouse (OW)*
- the *Multidimensional Warehouse (MDW)*

The Operational Warehouse can be further divided into the *Operational Warehouse - Staging (OWS)* and the *Operational Warehouse - Enriched (OWE).*

Each warehouse structure has its own set of specialized target warehouse tables that are unique to that structure. For example, the Operational Warehouse - Enriched (OWE) structure stores enriched data that is arranged in a normalized format to promote complex analytics. And the Multidimensional Warehouse (MDW) structure stores data that is arranged in a denormalized format (dimensional schema) for enhanced reporting capabilities.

The following graphic illustrates the various components comprising the EPM architecture and how each component relates to the others, including shared components which act as the foundation for both the EPM Warehouses and Analytical Applications.



EPM architecture

The dual data warehouse architecture helps to:

- Isolate and channel specific source data to the appropriate data warehouse structure for individual enrichment and modeling.

  PeopleSoft provides *extract, transform, and load (ETL) jobs* to extract information contained in your source systems, load it into the Operational Warehouse - Staging (OWS) structure, and migrate that data to the Operational Warehouse - Enriched (OWE) and the Multidimensional Warehouse (MDW) structures. And because the warehouse structures are logically separated, the ETL jobs can isolate and channel specific source data to the OWE or the MDW.

- Facilitate specialized, or tailored, data enrichment for your source data.

  PeopleSoft provides *EPM Foundation tools and processes* (a set of specialized tools, processes, and metadata) that prepare and enrich your source data for the EPM Warehouses and Analytical Applications.

The delivered target warehouse tables, ETL jobs, Foundation tools, and other packaged content work together to provide the underlying infrastructure on which the EPM Warehouses and Analytical Applications are built. Detailed information regarding the OWS, OWE, MDW, and EPM Foundation tools can be found in this chapter.

Note. EPM *data warehouse structures* refer to the OWS, OWE, and MDW, whereas *EPM Warehouses* refer to the PeopleSoft packaged warehouse solutions available for licensing, such as the Campus Solutions Warehouse and the Human Capital Management Warehouse.

Definition of a Data Warehouse

A textbook definition of a data warehouse is: a copy of transaction data specifically structured for query and analysis.

Transactional database applications have been widely used by the corporate world for over 30 years. Although data has been entered into dedicated transaction applications for decades, it has become apparent that extracting data from these systems for analytic purposes can be cumbersome and difficult.

Data warehousing is the process of taking data from legacy and transaction database systems and transforming it into organized information in a user-friendly format to encourage data analysis and support fact-based business decision-making.

A data warehouse is a central, integrated database that contains data from one or more operational sources and archive systems in an organization. It contains a copy of transaction data that is specifically structured for query analysis.

The mission of the data warehouse is to publish an organization's data assets to most effectively support decision-making. Because the data warehouse is a decision-support system, the main criterion of success is whether the data warehouse contributes to the most important decision-making processes in the organization.

See Also

Chapter 2, "Understanding PeopleSoft Enterprise Performance Management," Operational Warehouse - Staging (OWS), page 9

Chapter 2, "Understanding PeopleSoft Enterprise Performance Management," Multidimensional Warehouse (MDW), page 19

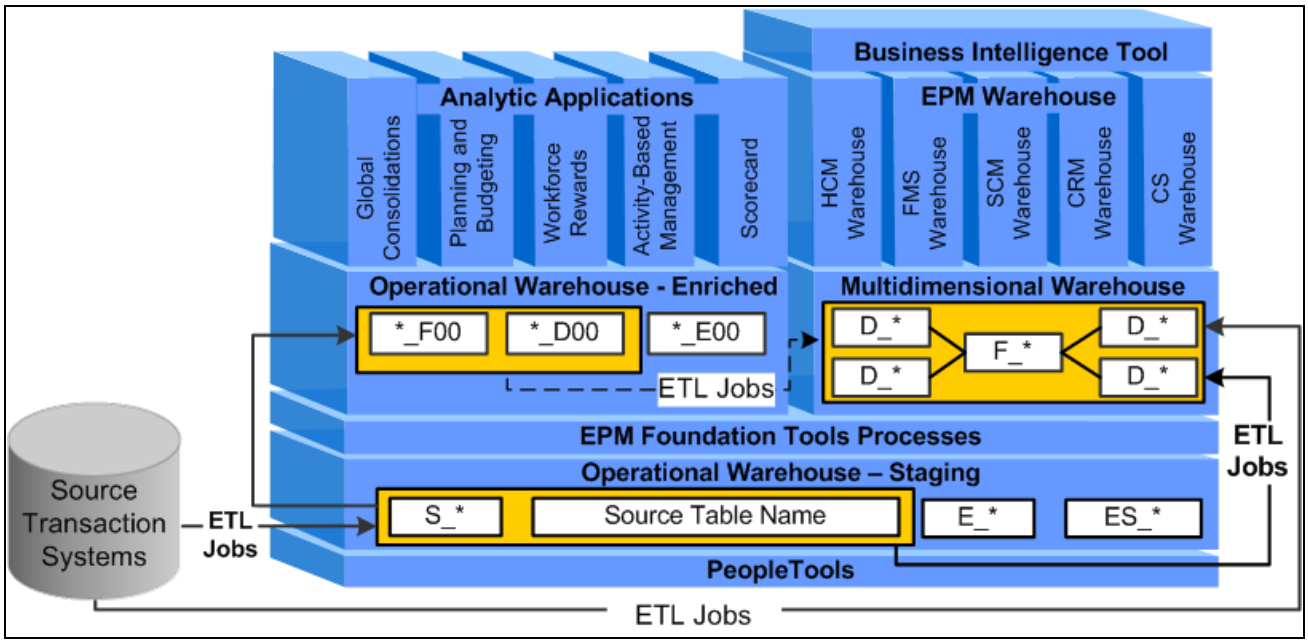
Chapter 2, "Understanding PeopleSoft Enterprise Performance Management," EPM Foundation Toolset, page 25

Extract, Transform, and Load (ETL) in EPM

PeopleSoft has an original equipment manufacturer (OEM) agreement with IBM WebSphere to supply *extract, transform, and load (ETL)* technology that supports source data acquisition and data movement within EPM . The ETL tool, IBM WebSphere *DataStage*, is delivered with EPM.

PeopleSoft uses IBM WebSphere DataStage to deliver prepackaged ETL jobs that extract information contained in PeopleSoft source systems, load it into the Operational Warehouse - Staging (OWS), and migrate that data to the Operational Warehouse - Enriched (OWE) and the Multidimensional Warehouse (MDW). But ETL jobs do more than migrate data; they also identify data for extraction and ensure the consistency and validity of your data. Because the ETL jobs are so versatile, separate tools and engines that extract, stage, and move data are not necessary.

The following graphic illustrates the various components comprising the EPM architecture and how data flows from source systems to the Operational and Multidimensional warehouses via the ETL process.



ETL process in EPM

As depicted in the diagram, source transaction data is extracted into OWS tables and migrated across warehouse layers using the aforementioned ETL jobs. Also, source data is sometimes extracted directly into the MDW.

You can use IBM WebSphere DataStage to build custom jobs for mapping your data into EPM . However, PeopleSoft does not support custom jobs.

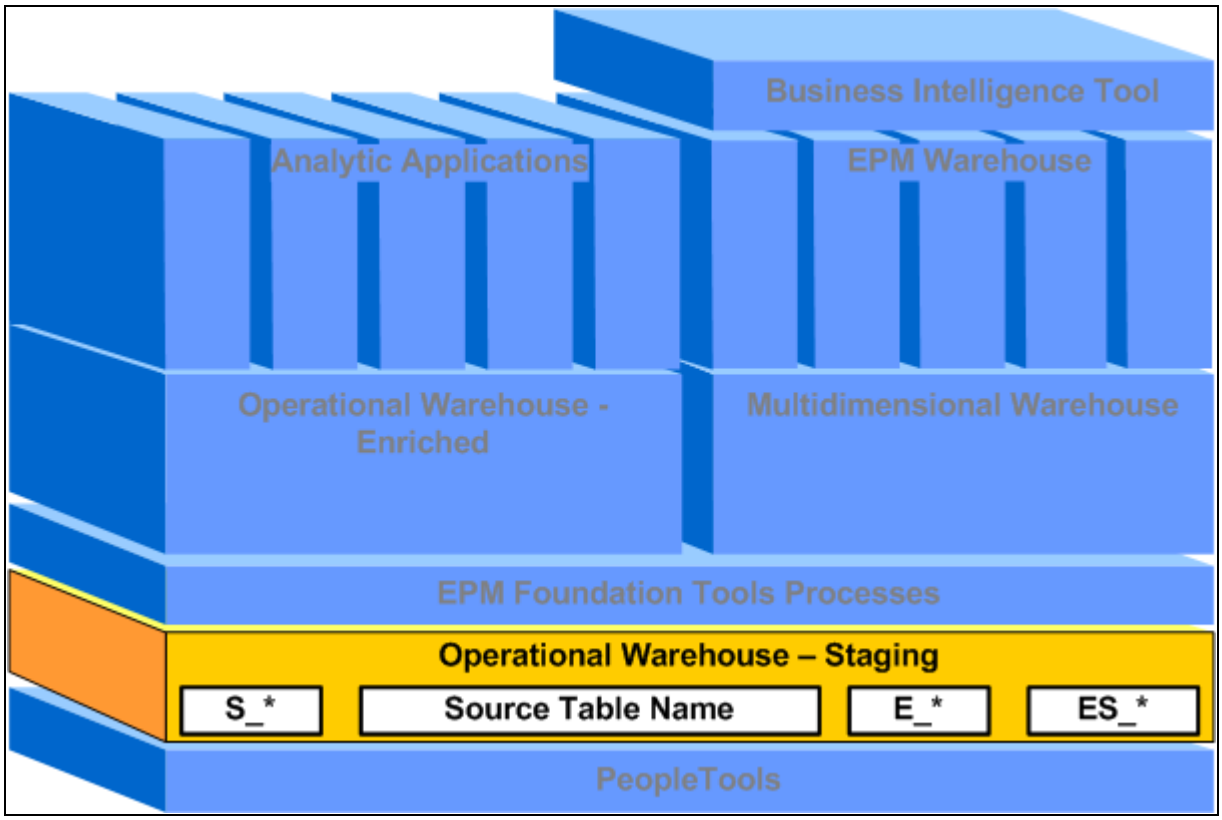
Detailed information regarding the ETL process can be found in the ETL section of this PeopleBook.

See Chapter 7, "Preparing to Load Source Data Into EPM," page 161.

Operational Warehouse - Staging (OWS)

The OWS structure is one of two subcomponents that comprise the Operational Warehouse. The OWS acts as an entry-point for your source transaction data into EPM and can house data from one or more of your PeopleSoft, legacy, or external source systems. The main function of the OWS is to provide a platform to offload, consolidate, and stage your source transaction data in preparation for enrichment.

The following graphic illustrates the OWS component of the EPM architecture and the target tables that are present in the OWS.



Operational Warehouse - Staging (OWS)

Source data is extracted into the OWS using prepackaged ETL jobs and loaded into target staging tables. No transformations are performed on your source data during this process and the system maintains the same source-level of granularity for your data. Source tables are extracted into the OWS, including all logically related tables, to ensure your source system data is semantically complete. For example, a table extracted into the OWS may have an associated related language table in the source system. The related language data from the associated table is also extracted into the OWS to maintain completeness and data integrity. Data stored in the OWS is used as input for the Operational Warehouse - Enriched (OWE) and the Multidimensional Warehouse (MDW) structures.

Note. The OWS does not contain reporting tables nor prepackaged reports built on the core OWS target tables.

OWS Core Target Tables

OWS core target tables contain data extracted from PeopleSoft Enterprise source systems. OWS target tables are permanent tables (as opposed to temporary tables), and can store historical data. However, it is not the recommended location for historical data as the tables can be purged from time to time depending on your operational needs. The structure of the OWS target tables match the structure of the source transaction tables with the addition of a source system identification column (SRC_SYS_ID) , which enables you to track the origin of your data.

Note. Certain OWS target tables have specific non-key columns that can be "activated" as key columns if your business requirements necessitate it.

Sample OWS Target Table

The following is a sample OWS target table page shown in Application Designer.

	Num	Field Name	Type	Len	Format	Short Name	Long Name
	1	SETID	Char	5	Upper	SetID	SetID
	2	ABSENCE_CLASS	Char	4	Upper	Absence Class	Absence Class
	3	SRC_SYS_ID	Char	5	Upper	Source ID	Source System Identifica
	4	DESCR	Char	30	Mixed	Descr	Description
	5	DESCRSHORT	Char	10	Mixed	Short Desc	Short Description
	6	LOAD_OWS_SBR	SRec				

OWS target table - ABS_CLASS_TBL

OWS Target Table Naming Convention

OWS target tables use the following naming conventions:

- S_[*source table name*]
- [*source table name*]

OWS Error Tables

The OWS contains error tables used in the data validation process. The data validation process uses ETL jobs to verify the integrity and completeness of the data entering OWE and MDW target tables. The validation process can perform dimension key validation (for example, verifying that customer ID fact value has a corresponding customer ID dimension value) and general key validation (for example, verifying the pre-fact customer ID in the OWS table has a corresponding customer ID in the OWE or MDW table), as well as ensure source business unit and setID are properly mapped to EPM values and source codes are properly mapped to EPM code values.

Data failing the validation process are sent to OWS error tables. It is important to note that the OWS error tables have a different structure than the error tables in the OWE and perform a very different function. The OWS error table mirrors the key structure and other columns of its corresponding data table and has additional fields to facilitate troubleshooting. The following OWS error table columns represent some of the columns provided for troubleshooting:

- LOAD_OWS_SBR: The values for these columns are copied from the failing data row. The reason for copying values from the failed data row is that it provides vital load information such as batchID and load timestamp for the data row.
- Target Table: This column lists the target table for the job.
- Failed data source table and column name: The source table and column from which the failing data originated. Knowing the name of the failed source data table is especially useful when the job loading the failed data contains a multi-source-table join.
- Failed Data Value: The actual value that failed validation.
- Lookup table and column name: The table and column against which the failed lookup was performed.

Detailed information regarding the data validation process can be found in the ETL section of this PeopleBook.

See Chapter 7, "Preparing to Load Source Data Into EPM," page 161.

Sample OWS Error Table

The following is a sample OWS error table page shown in Application Designer.

	Num	Field Name	Type	Len	Format	Short Name	Long Name
	1	ERROR_SID	Nbr	10	Raw B	Error Row SID	Error Row SID
	2	SETID	Char	5	Upper	SetID	SetID
	3	ABSENCE_CLASS	Char	4	Upper	Absence Class	Absence Class
	4	SRC_SYS_ID	Char	5	Upper	Source ID	Source System Identification
	5	ERR_TRACE_SBR	SRec				
	6	LOAD_OWS_SBR	SRec				

OWS error table - E_ABS_CLASS_TBL

OWS Error Table Naming Convention

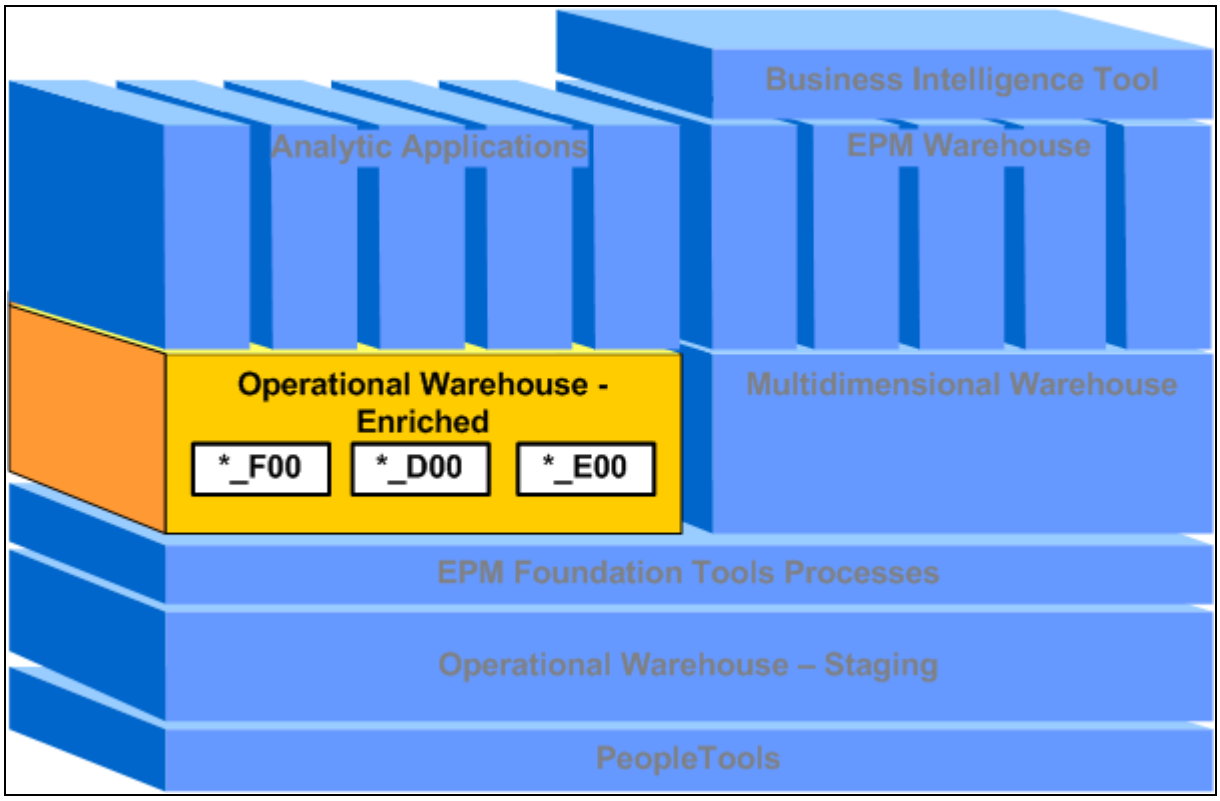
OWS error tables use the following naming conventions:

- PS_E_[*OWS table name*]
- PS_ES_[*OWS table name*]

Operational Warehouse - Enriched (OWE)

The OWE structure is the second of two subcomponents that comprise the Operational Warehouse. The OWE stores enriched data that is arranged in a normalized format and mapped to warehouse business units (WBU). Enrichment can entail many transformations to your data, including (but not limited to) conversion to a common currency, common calendar, or a common ledger, or aggregating data to a common warehouse business unit. The PeopleSoft Analytical Applications use the enriched data in the OWE to perform analysis and reporting.

The following graphic illustrates the OWE component of the EPM architecture and the target tables that are present in the OWE.



Operational Warehouse - Enriched (OWE)

Data is extracted into the OWE using prepackaged ETL jobs and loaded into target dimension (D00) and fact (F00) tables. The structure of these tables are quite different from the OWS tables because they are arranged in a normalized format and organized data around warehouse business units. In addition, OWE tables are augmented with subrecords which help facilitate the ETL process and tracking data lineage. OWE tables store data permanently and can maintain history (as opposed to temporary tables which remove data at the end of an ETL job).

Tools and Processes Associated with the OWE

EPM is delivered with several tools and processes that enable you to enrich and manage the data stored in the OWE. The following are some of the tools and processes used only with the OWE:

- Performance ledger template setup.
- Detail ledger setup.
- Model and scenario setup.
- Roll-up processing.
- Profit manager.
- EPM object auditing.
- Mass validate processing.
- Mass compile processing.

- Tree utility setup.
- Data manager processing.
- Allocation manager processing.

OWE Dimension (D00) Tables

An OWE dimension table provides additional attributes about a fact for greater flexibility in reporting. Dimensions are derived from operational applications and are cleansed and transformed during data migration. Examples of dimension tables include: product, customer, channel, department, personal data, and accounts. Some of the fields associated with an OWE dimension table are:

- SetID: Key column.
- DIMENSION_ID: Key column.
- EFFDT: This is the same date as the source. If an EFFDT or an alternative date, such as a date time stamp, does not exist, the system creates one and sets it to the date the dimension data is loaded.
- EFF_STATUS.
- KEY Fields from the source table.
- SET CONTROL FIELD. This is BUSINESS_UNIT.
- RELATED LANGUAGE RECORD. This is the same with an extension of _LNG.

Sample OWE Dimension Table

The following is a sample OWE dimension table page shown in Application Designer.

Num	Field Name	Type	Len	Format	Short Name	Long Name
1	**SETID**	Char	5	Upper	SetID	SetID
2	CUST_ID	Char	15	Upper	Customer	Customer ID
3	EFFDT	Date	10		Eff Date	Effective Date
4	EFF_STATUS	Char	1	Upper	Status	Status as of Effective D
5	NAMESHORT	Char	10	Mixed	Short Name	Short Name
6	CUSTOMER_TYPE	Char	1	Upper	Type	Customer Type
7	CUSTOMER_GROUP	Char	10	Upper	Cust Group	Customer Group
8	SINCE_DT	Date	10		Since	Customer Since
9	ADD_DT	Date	10		Date Added	Customer Added On
10	NAME1	Char	50	Mixed	Name	Name 1
11	NAME2	Char	40	Mixed	Name 2	Name 2
12	NAME3	Char	40	Mixed	Name 3	Name 3
13	SALES_PERSON	Char	8	Upper	Sales	Sales Person
14	CORPORATE_SETID	Char	5	Upper	SetID	SetID
15	**CORPORATE_CUST_ID**	Char	15	Upper	Corporate	Corporate Customer
16	VENDOR_SETID	Char	5	Upper	Vndr SetID	Vendor SetID
17	VENDOR_ID	Char	10	Upper	Vendor	Vendor ID
18	FORMER_NAME_1	Char	40	Mixed	Former Name	Former Name
19	CURRENCY_CD	Char	3	Upper	Currency	Currency Code
20	SHIP_TO_FLG	Char	1	Upper	Ship To	Ship To Customer
21	BILL_TO_FLG	Char	1	Upper	Bill To	Bill To Customer
22	SOLD_TO_FLG	Char	1	Upper	Sold To	Sold To Customer
23	CUST_LEVEL	Char	1	Upper	Level	Customer Level
24	ADDRESS_SEQ_NUM	Nbr	3		Addr	Address Sequence Num
25	ADDRESS_SEQ_SOLD	Nbr	3		Address	Primary Address - Sold T
26	ADDRESS_SEQ_SHIP	Nbr	3		Address	Primary Address - Ship T
27	TAXPAYER_ID	Char	14	Upper	Tax ID	Taxpayer ID
28	COMPANY_SIZE	Nbr	10		Company Size	Company Size
29	INDUSTRY_ID	Char	30	Upper	CRM Industry ID	CRM Industry ID
30	BO_ID	Sign	31		BO ID	Business Object ID
31	BO_ID_PARENT	Nbr	31		Parent	Parent Company
32	SOURCE_IND	Char	4	Upper	Source?	Source Indicator
33	PROCESS_INSTANCE	Nbr	10		Instance	Process Instance
34	LOAD_OWE_SBR	SRec				

OWE dimension - CUSTOMER_D00

OWE Dimension Table Naming Convention

OWE dimension tables use the following naming convention, [*table name*]_D00

OWE Fact (F00) Tables

An OWE fact table contains measures (from across the enterprise) for analyzing performance. Some of the fields associated with an OWE fact table are:

- BUSINESS_UNIT: This field enables the fact data to be shared across different dimensions, as they are based on SetIDs.
- FACT KEY.
- ASOF_DT: This is for non-cumulative facts (for example, account balance).
- PF_TRANS_DT: This is for cumulative facts (for example, billing transactions).

- All KEY Fields: These are required for uniqueness.
- DEFAULT VALUES include:
  - BUSINESS_UNIT: This will have a default table set to OPR_DEF_TBL_FS and a default field set to BUSINESS_UNIT.
  - EFFDT: This will have a default set to %DATE.
  - EFF_STATUS: This will have a default set to *A.*
  - TRANSLATE VALUES: These values, if any exist, must be set to the XLATTABLE.
- PF_TRANS_DT: This is set to the source record's transaction date. In addition, the source transaction date field is included in the data warehouse fact table.

Sample OWE Fact Table

The following is a sample OWE fact table page shown in Application Designer.

Num	Field Name	Type	Len	Format	Short Name	Long Name
1	BUSINESS_UNIT	Char	5	Upper	Unit	Business Unit
2	CUST_ID	Char	15	Upper	Customer	Customer ID
3	EFFDT	Date	10		Eff Date	Effective Date
4	BAL_AMT	Sign	23.3		Balance	Customer Balance
5	HI_BAL_AMT	Sign	23.3		Hi Balance	Hi Balance
6	HI_BAL_DT	Date	10		Hi Balance	Hi Balance Date
7	ITEM	Char	30	Upper	Item ID	Item ID
8	ITEM_AMT	Sign	23.3		Amount	Amount
9	ITEM_DT	Date	10		Date	Item Date
10	DEPOSIT_BU	Char	5	Upper	Unit	Deposit Unit
11	DEPOSIT_ID	Char	15	Upper	Deposit ID	Deposit ID
12	PAYMENT_SEQ_NUM	Nbr	6		Seq	Payment Sequence
13	PAYMENT_DT	Date	10		Date	Payment Date
14	POST_DT	Date	10		Posted	Posted Date
15	POSTED_PI	Nbr	10		Proc Inst	Last Post Process Instan
16	AGED_DT	Date	10		Aged	Aged Date
17	AGED_PI	Nbr	10		Proc Inst	Last Aging Process Insta
18	PAYMENT_ID	Char	15	Upper	Payment ID	Payment ID
19	PAYMENT_AMT	Sign	23.3		Amount	Payment Amount
20	ENTRY_CURRENCY	Char	3	Upper	Ent Crncy	Entry Currency Code
21	PAYMENT_CURRENCY	Char	3	Upper	Currency	Payment Currency
22	CURRENCY_CD	Char	3	Upper	Currency	Currency Code
23	AGED_AMT_CR_CHK	Sign	23.3		Amt Checkd	Amount Credit Checked
24	ITEM_LINE	Nbr	6		Line	Item Line
25	PROCESS_INSTANCE	Nbr	10		Instance	Process Instance

OWE fact - ACCT_REC_F00

OWE Fact Table Naming Convention

OWE fact tables use the following naming convention, [*table name*]_F00

OWE Temporary Tables

OWE temporary tables support parallel processing. EPM is delivered with three sets of temporary tables. You can define additional sets of tables when needed.

The project EPM_TEMP_TABLES contains one instance of every temporary table, enabling you to create new temporary table suites, if necessary. A temporary table layout and key structure differs from its respective fact or dimension data warehouse table in that the organizational unit (setID or business unit) and the effective date are not keys.

Note. If you must create more temporary tables than the ones delivered with PeopleSoft EPM, see the delivered project, EPM_TEMP_TABLES. It contains one instance of every temporary table, enabling you to create new temporary table suites, if necessary.

See Chapter 20, "Streamlining Processing with Jobstreams," Creating Additional Instances of Temporary Tables for Record Suites, page 481.

OWE Temporary Table Naming Convention

OWE temporary tables use the following naming convention, [*table name*]_T

Specialized Reporting Tables

The OWE features tables that have been designed specifically to enhance reporting capabilities. Those tables are the performance ledger table (PF_LEDGER_F00), performance journal table (PF_JRNL_F00), and the performance statistics table (PF_STAT_F00). Creating specialized tables in this manner enables you to move away from storing all of your accounting data in your general ledger to make your general ledger perform as it should—as a method for compliance reporting *only*.

The performance journal and performance ledger tables are described in more detail later in this PeopleBook.

Performance Ledger Table

The performance ledger table (PF_LEDGER_F00) is a central fact table within EPM. The performance ledger table is an accumulation of monetary amount facts over a period of time. The primary function of the performance ledger table is to support PeopleSoft EPM reporting. The PF_LEDGER_F00 is the source for one of the data marts.

Note. The performance ledger table should not be confused with a general ledger from an online transaction processing (OLTP) system. The performance ledger contains all information mapped from a general ledger and enriched through one (or more) of the PeopleSoft EPM engines.

Information that has been processed through an PeopleSoft EPM engine, for instance the ABM engine or Data Manager, is stored in a temporary performance journal staging table (PF_JRNL_T).

The PF Edit engine enables you to verify the data in the temporary journal table and moves valid data to the final table, the PF_JRNL_F00. Errors are placed in the PF_JRNL_E00, the error table for the journal table. The PF Post takes the detailed information from the performance journal table, aggregates it to the desired level of summarization and posts it to the PF_LEDGER_F00 for reporting.

PeopleSoft EPM reporting tools support multidimensional analysis based primarily on profitability dimensions such as customer, product, and channel. You can use one, two, or more of these dimensions within your models, or configure the application to add more dimensions, or change the existing ones. No matter which dimensions you select, however, you need to consider how to populate the performance ledger table with meaningful multidimensional data.

Performance Journal Table

The performance journal table (PF_JRNL_F00):

- Contains data that is not yet summarized.
- Is a fact table, or multiple fact tables, within EPM.
- Is a collection of batches of amount facts staged for validation and posting to the performance ledger table.
- Supports drill down from reports produced against the performance ledger table.

The PF Edit engines moves data to the performance journal fact table. The PF Post process accumulates valid transactions from the performance journal table, and inserts summarized rows into the performance ledger. There is a "many to one" relationship between the performance journal and the performance ledger tables.

Performance Statistics Table

The performance statistics table (PF_STAT_F00) is similar to the performance ledger table (PF_LEDGER_F00) in its layout.

OWE Error Tables (for Profit Manager only)

The OWE contains error tables used to identify flawed data in certain OWE target tables. There are a small number of delivered OWE error tables and they are used only for Profit Manager. Profit Manager uses specific business rules to validate and format data in its related OWE target tables. If the business rules are not met, then the flawed records are written to an OWE error table and a message describing the error is written to a detail error message table (TSE table). If your load results in errors, you can use PF Modification to correct the errors. You can correct the errors using the PeopleSoft Application Designer and then migrate the corrected tables to the target. The following OWE error tables are delivered:

- BP_LED_BUDG_E00
- BP_LED_E00
- BP_LED_KK_E00
- BP_LED_PROJ_E00
- GC_JRNL_MGT_E00
- LEDGER_E00
- PF_JRNL_E00

See Chapter 21, "Setting Up and Using Profit Manager," page 497.

Sample OWE Error Table

The following is a sample OWE error table page shown in Application Designer.

Num	Field Name	Type	Len	Format	Short Name	Long Name
1	BUSINESS_UNIT	Char	5	Upper	Unit	Business Unit
2	LEDGER	Char	10	Upper	Ledger	Ledger
3	FISCAL_YEAR	Nbr	4		Year	Fiscal Year
4	ACCOUNTING_PERIOD	Nbr	3		Period	Accounting Period
5	ACCOUNT	Char	10	Upper	Acct	Account
6	ALTACCT	Char	10	Upper	Alt Acct	Alternate Account
7	STATISTICS_CODE	Char	3	Upper	Stat	Statistics Code
8	BP_CF9B_AK_SBR	SRec				
9	PROJECT_ID	Char	15	Upper	Project	Project
10	CURRENCY_CD	Char	3	Upper	Currency	Currency Code
11	BOOK_CODE	Char	4	Upper	Book Code	Book Code
12	GL_ADJUST_TYPE	Char	4	Upper	Adjustment	Adjustment Type
13	**PF_EDIT_SEQ_NUM**	Nbr	6		Last Edit Seq #	Last Edit Seq Number
14	POSTED_TOTAL_AMT	Sign	23.3		Total Amt	Posted Total Amount
15	POSTED_BASE_AMT	Sign	23.3		Amount	Posted Base Currency A
16	POSTED_TRAN_AMT	Sign	23.3		Transaction Amt	Posted Transaction Amo
17	BASE_CURRENCY	Char	3	Upper	Base Curr	Base Currency
18	DTTM_STAMP_SEC	DtTm	26	Scnds	DateTime	Last Update DateTime
19	ERROR_FLAG	Char	1	Upper	Err Flag	Error Flag
20	PROCESS_INSTANCE	Nbr	10		Instance	Process Instance

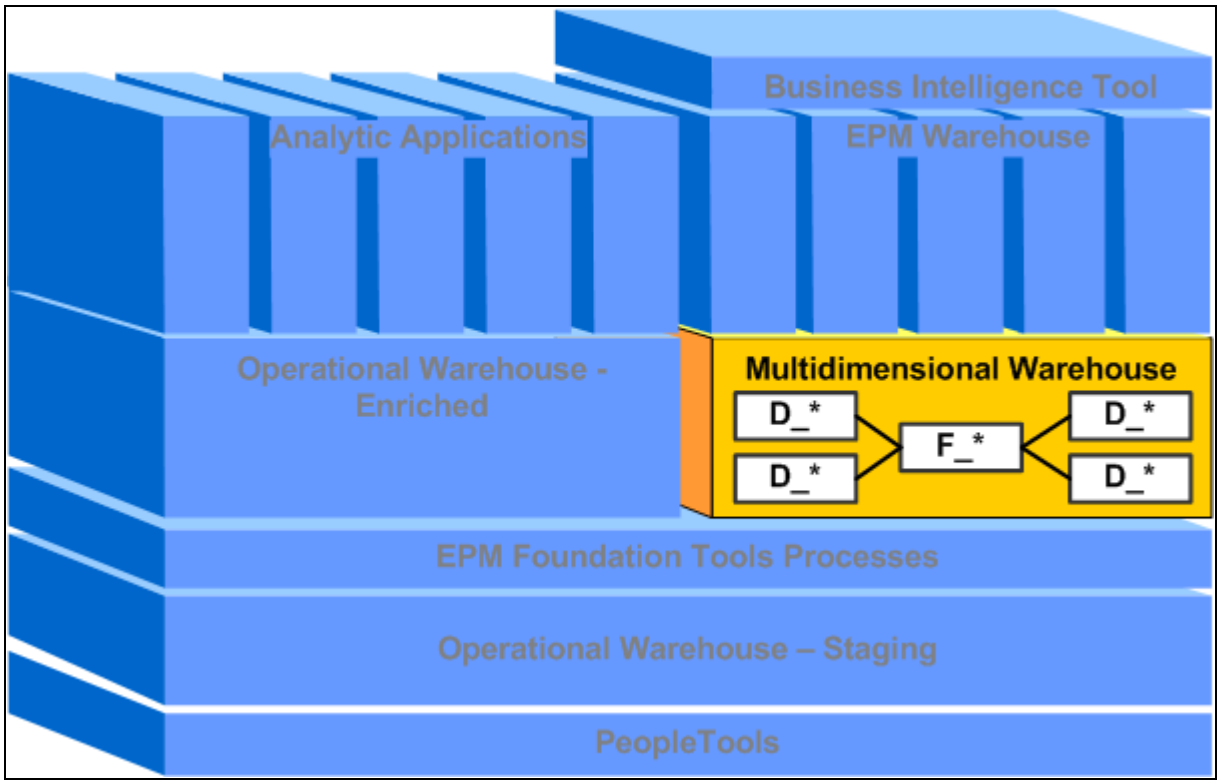
OWE error table - BP_LED_E00

OWE Error Table Naming Convention

OWE error tables use the following naming convention, [*table name*]_E00

Multidimensional Warehouse (MDW)

The Multidimensional Warehouse is the third data structure in EPM. The following graphic illustrates the MDW component of the EPM architecture and the target tables that are present in the MDW.



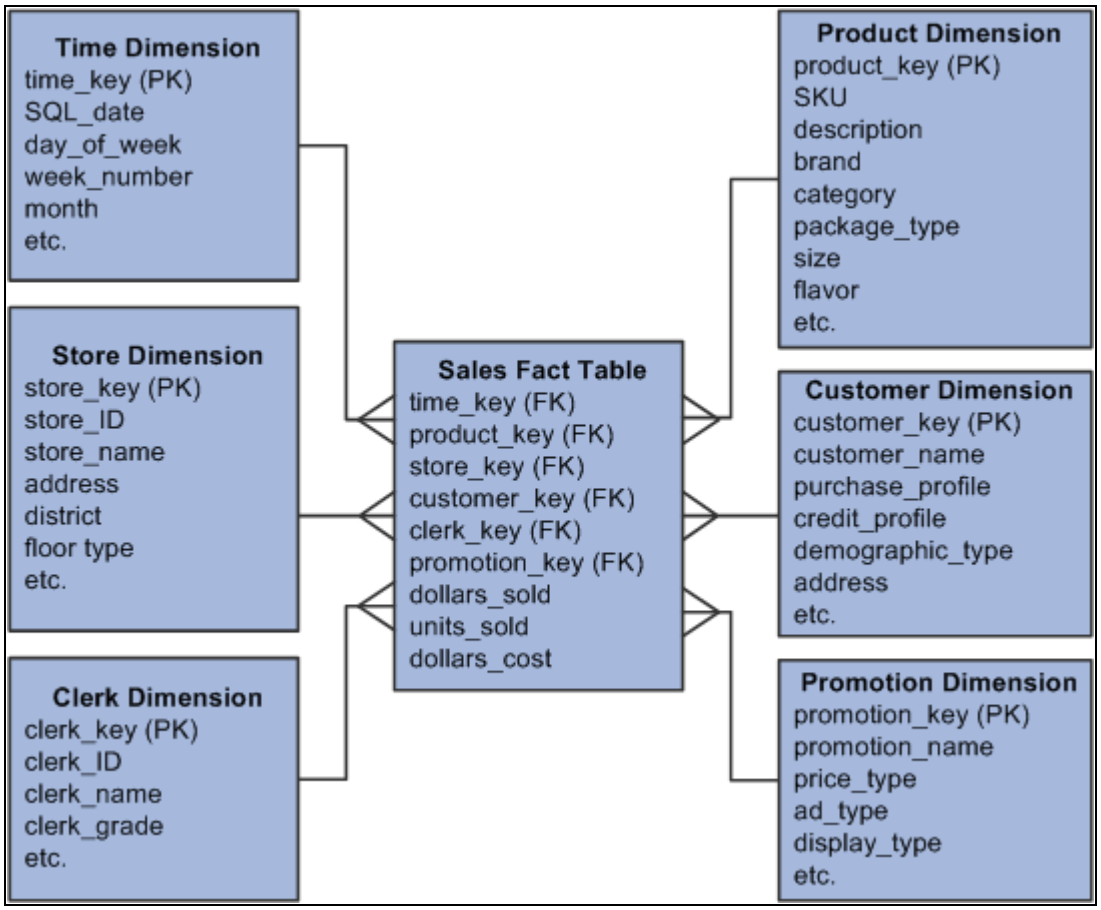
Multidimensional Warehouse (MDW)

The MDW stores dimensionalized data that is grouped into one or more business processes, better known as a *dimensional schema*, used for business intelligence and ad hoc reporting. The data is stored in a *star schema* (a fact table associated with a series of dimension tables) and generally contains data loaded from the OWS.

The star schema arrangement depends entirely on *primary key* and *foreign key* relationships. A primary key is a column (or columns) in a dimension table whose values uniquely identify each row in the table. Primary keys enforce entity integrity by uniquely identifying entity instances. A foreign key is a column or columns in a fact table whose values match the primary key values of a given dimension table. This way references can be made between a fact and dimension table. Foreign keys enforce referential integrity by completing an association between two entities.

Note. MDW dimensions use a *surrogate key*, a unique key generated from production keys by the ETL process. The surrogate key is not derived from any data in the EPM database and acts as the primary key in a MDW dimension. See the next section for more information on surrogate keys in the MDW.

The following graphic provides an example of a star schema and its primary and foreign key relationships:



Dimensional Model Example

Although data loaded into the MDW is primarily derived from the OWS, there are exceptions to this rule. Profitability and Global Consolidations data for the Financial Management Solutions (FMS) Warehouse is loaded into the MDW from the OWE.

External survey data for the HCM Warehouse is loaded into the MDW from the OWE.

Online Marketing data is loaded into the MDW directly from the source system, and bypasses the Operational Warehouse entirely.

Surrogate Keys

Surrogate keys provide a means of defining unique keys whose values, with the exception of the Time and Calendar dimensions, are anonymous—that is, the value of a surrogate key has no significance to the application using it and is strictly an artificial value. The system uses surrogate keys specifically as a means of joining structures. To speed up query access, the MDW resolves PeopleSoft-specific programming constructs, such as SetIDs and effective dates and replaces them with surrogate IDs as key columns. Surrogate keys have no relationship to the business or production key. Surrogate keys are present in dimension tables as the primary key and in fact tables as foreign keys to dimensions. However, the dimension record retains the business key as an alternate-key attribute. Surrogate keys are four-byte integers and their size does not change even when production key changes in size.

Although surrogate keys usually do not have any "intelligence," that is, their value has no meaning, in certain situations, such as the Gregorian Calendar and Time dimensions, intelligent surrogate keys are used. These intelligent keys enable the ETL process to run more quickly by providing the option of avoiding a lookup on corresponding dimensions.

Surrogate key fields usually have the suffix *_SID* (Surrogate ID).

Surrogate Keys and the ETL Process

Surrogate keys are generated from production keys using the DataStage routine *KeyMgtNextValueConcurent()*, which receives an input parameter and a name identifying the sequence. The surrogate key can be unique per single dimension target (D) or unique across the whole (W) multidimensional warehouse. This process is enabled by the environment parameter named SID_UNIQUENESS. The value for this parameter is provided at run time. If the value is *D*, then this routine is called with a dimension job name for which a surrogate key must be assigned and it returns the next available number. If not, the routine is called with *EPM* as the sequence identifier.

You do not have to take any action to create surrogate keys; they are generated during the ETL process within the aforementioned DataStage routine. The DataStage routine retrieves the next surrogate key value and assigns it to the surrogate key that it is currently creating. When the ETL process copies a dimension row from the source system into the MDW, the ETL process performs a lookup on the dimension table. If the dimension row (with same business keys) does not exist in the dimension table, the process inserts a row with a new surrogate key value. If the dimension row already exists in the dimension table, the process updates the existing row with the incoming row value. When the ETL process copies a fact row from the source system into the MDW, for each dimension key in the fact row, the system performs a lookup on the dimension table and retrieves the corresponding surrogate key value. This surrogate key is the foreign key value in the fact row in the MDW. If the system does not locate a dimension value in the fact row in the dimension table, that is a data exception and an error results.

Surrogate Key Benefits

Surrogate keys provide benefits such as:

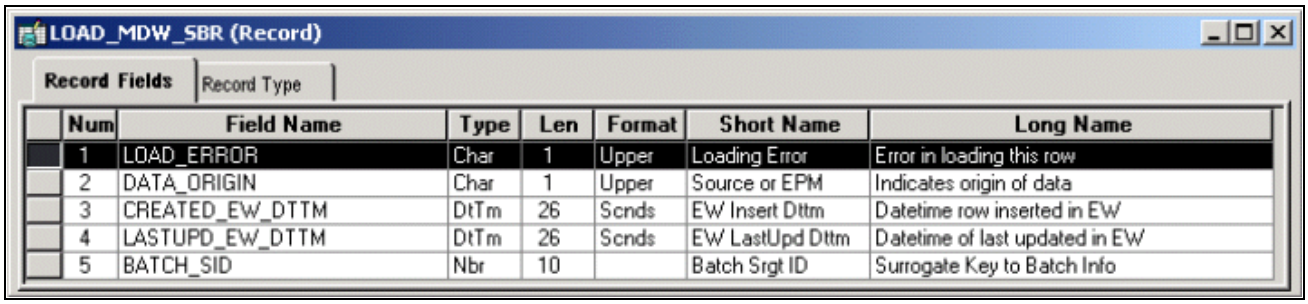
- The ability to easily and structurally conform a dimension when being sourced from multiple systems.
- Disassociation from operational system changes.

  Because surrogate key generation is controlled by the warehouse, it is not influenced by operational system changes.
- The ability to handle unspecified or missing key values.
- A graceful mechanism to handle changes in history.

  Multiple versions of a dimension can be maintained with different surrogate (primary) keys, yet with the same business (identifying) key.
- Performance enhancement of queries, because a surrogate key is a single column numeric key, thus the joins using surrogate keys are faster than ones using multi-column business keys.

Audit Fields

Audit fields track extract, transform, and load (ETL) loading information, such as when the row was loaded or last modified or the batch in which the row was loaded. This information is included in a subrecord. The subrecord added to MDW tables is called LOAD_MDW_SBR. Subrecords are always added at the end of a record; no fields exist after this subrecord in any table. The following example shows a typical LOAD_MDW_SBR subrecord.


LOAD_MDW_SBR (Record)

Record Fields | Record Type

Num	Field Name	Type	Len	Format	Short Name	Long Name
1	LOAD_ERROR	Char	1	Upper	Loading Error	Error in loading this row
2	DATA_ORIGIN	Char	1	Upper	Source or EPM	Indicates origin of data
3	CREATED_EW_DTTM	DtTm	26	Scnds	EW Insert Dttm	Datetime row inserted in EW
4	LASTUPD_EW_DTTM	DtTm	26	Scnds	EW LastUpd Dttm	Datetime of last updated in EW
5	BATCH_SID	Nbr	10		Batch Srgt ID	Surrogate Key to Batch Info

LOAD_MDW_SBR record example

Data Aggregation

Tables in the MDW contain source data at the same granularity as the source system. Required data aggregation is carried out at run time by the business intelligence tool. This allows for better control of aggregation strategies by the business intelligence tool, because aggregation requirements vary from customer to customer.

MDW Dimension Tables

Dimension tables contain surrogate keys as the primary key and are a single column key containing only the surrogate key column. Surrogate keys usually have *_SID* (surrogate ID) appended to the field name. Dimension tables retain source system business key fields as non-key attribute columns in the dimension table. However, these are not used for joins with fact tables. For example, in the Customer dimension, the original business key field CUST_ID is retained, if it exists in the source table, but is no longer included in the key. The SetID is also retained, if it exists in the source table, as a nonkey attribute; the value contained in the SetID is the same as in the source system.

If a dimension is SetID-based, the MDW table contains the source SetID and the performance (PF) SetID, which is named *SETID*.

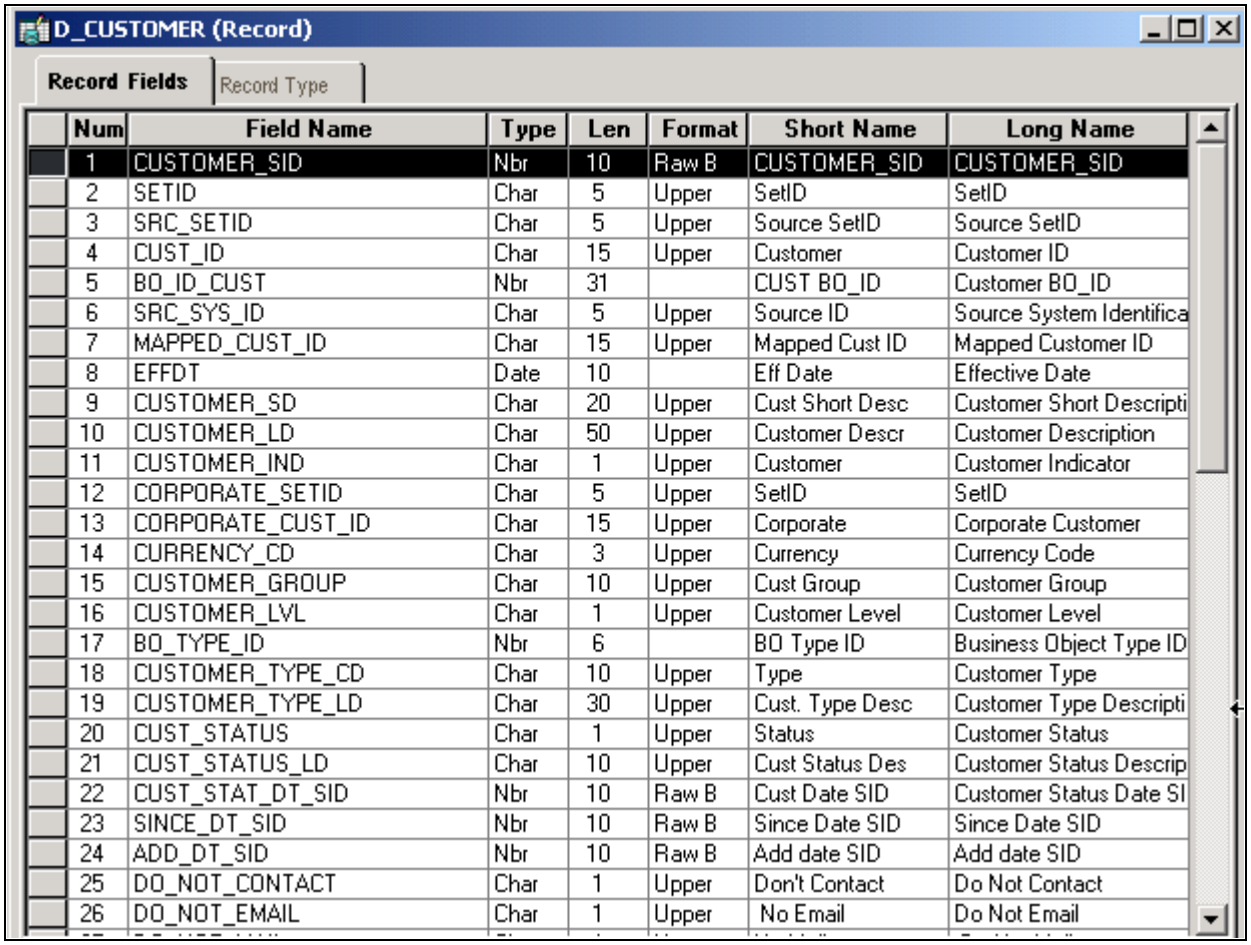
If a dimension contains a description text, a related language table is often defined for this dimension. The ETL process populates this table if a customer requires multilanguage processing. The key for this table is the surrogate key ID, plus the language code field, *LANGUAGE_CD*, which contains the code for the additional language.

Note. You can find more information about multilanguage processing for the multidimensional warehouse in your EPM Warehouse specific PeopleBook (for example, the *PeopleSoft Enterprise Campus Solutions Warehouse PeopleBook*).

Shared Dimensions

Dimensions such as Account, Customer, Department, or Person are examples of shared dimensions. Shared dimensions are either exactly the same—including key structure—or an exact subset of another dimension; that is, shared dimensions are structurally identical every place in which they are used. Shared dimensions are used across all EPM warehouse products, such as the Campus Solutions Warehouse and the Financial Management Solutions Warehouse.

When using a shared dimension, the system consistently interprets attributes; hence rollups across data marts are possible and consistent. When a warehouse is provided data from multiple sources, a shared dimension is typically (but not always) built from multiple source structures. The following is a sample MDW shared dimension shown in Application Designer.


D_CUSTOMER (Record)

Record Fields | Record Type

Num	Field Name	Type	Len	Format	Short Name	Long Name
1	CUSTOMER_SID	Nbr	10	Raw B	CUSTOMER_SID	CUSTOMER_SID
2	SETID	Char	5	Upper	SetID	SetID
3	SRC_SETID	Char	5	Upper	Source SetID	Source SetID
4	CUST_ID	Char	15	Upper	Customer	Customer ID
5	BO_ID_CUST	Nbr	31		CUST BO_ID	Customer BO_ID
6	SRC_SYS_ID	Char	5	Upper	Source ID	Source System Identifica
7	MAPPED_CUST_ID	Char	15	Upper	Mapped Cust ID	Mapped Customer ID
8	EFFDT	Date	10		Eff Date	Effective Date
9	CUSTOMER_SD	Char	20	Upper	Cust Short Desc	Customer Short Descripti
10	CUSTOMER_LD	Char	50	Upper	Customer Descr	Customer Description
11	CUSTOMER_IND	Char	1	Upper	Customer	Customer Indicator
12	CORPORATE_SETID	Char	5	Upper	SetID	SetID
13	CORPORATE_CUST_ID	Char	15	Upper	Corporate	Corporate Customer
14	CURRENCY_CD	Char	3	Upper	Currency	Currency Code
15	CUSTOMER_GROUP	Char	10	Upper	Cust Group	Customer Group
16	CUSTOMER_LVL	Char	1	Upper	Customer Level	Customer Level
17	BO_TYPE_ID	Nbr	6		BO Type ID	Business Object Type ID
18	CUSTOMER_TYPE_CD	Char	10	Upper	Type	Customer Type
19	CUSTOMER_TYPE_LD	Char	30	Upper	Cust. Type Desc	Customer Type Descripti
20	CUST_STATUS	Char	1	Upper	Status	Customer Status
21	CUST_STATUS_LD	Char	10	Upper	Cust Status Des	Customer Status Descrip
22	CUST_STAT_DT_SID	Nbr	10	Raw B	Cust Date SID	Customer Status Date SI
23	SINCE_DT_SID	Nbr	10	Raw B	Since Date SID	Since Date SID
24	ADD_DT_SID	Nbr	10	Raw B	Add date SID	Add date SID
25	DO_NOT_CONTACT	Char	1	Upper	Don't Contact	Do Not Contact
26	DO_NOT_EMAIL	Char	1	Upper	No Email	Do Not Email

EPM conformed dimension

MDW Dimension Table Naming Convention

MDW dimension tables use the following naming convention: D_[*table name*].

MDW Fact Tables

Fact tables foreign common key fields to dimensions. Dimension tables have a surrogate ID column that is the primary key of that dimension. A fact table may use these dimension surrogate IDs as foreign keys to the dimension table. In the dimensional model example graphic presented previously, the Sales fact table contains six foreign keys, each one matching a dimension surrounding the fact table.

Periodic Snapshot Fact Tables

Periodic Snapshots provide a view of the cumulative performance of the business at regular, predictable time intervals. Unlike a transaction fact table that loads a row of data for each event occurrence, the periodic snapshot fact table captures the event at the interval of a day, week, or month, and another capture at the interval of the next period, and so on. These periodic snapshots are stacked consecutively into the fact table. The periodic snapshot fact table often is the only place to easily retrieve a regular, predictable, trend view of the key business performance metrics.

Accumulating Fact Tables

Accumulating snapshots represent an indeterminate time span, covering the complete life of a transaction or discrete product. Accumulating snapshots almost always have multiple date stamps, representing the predictable major events or phases that take place during the course of a lifetime. Since many of these dates are not known when the fact row is first loaded, we must use surrogate date keys to handle undefined dates.

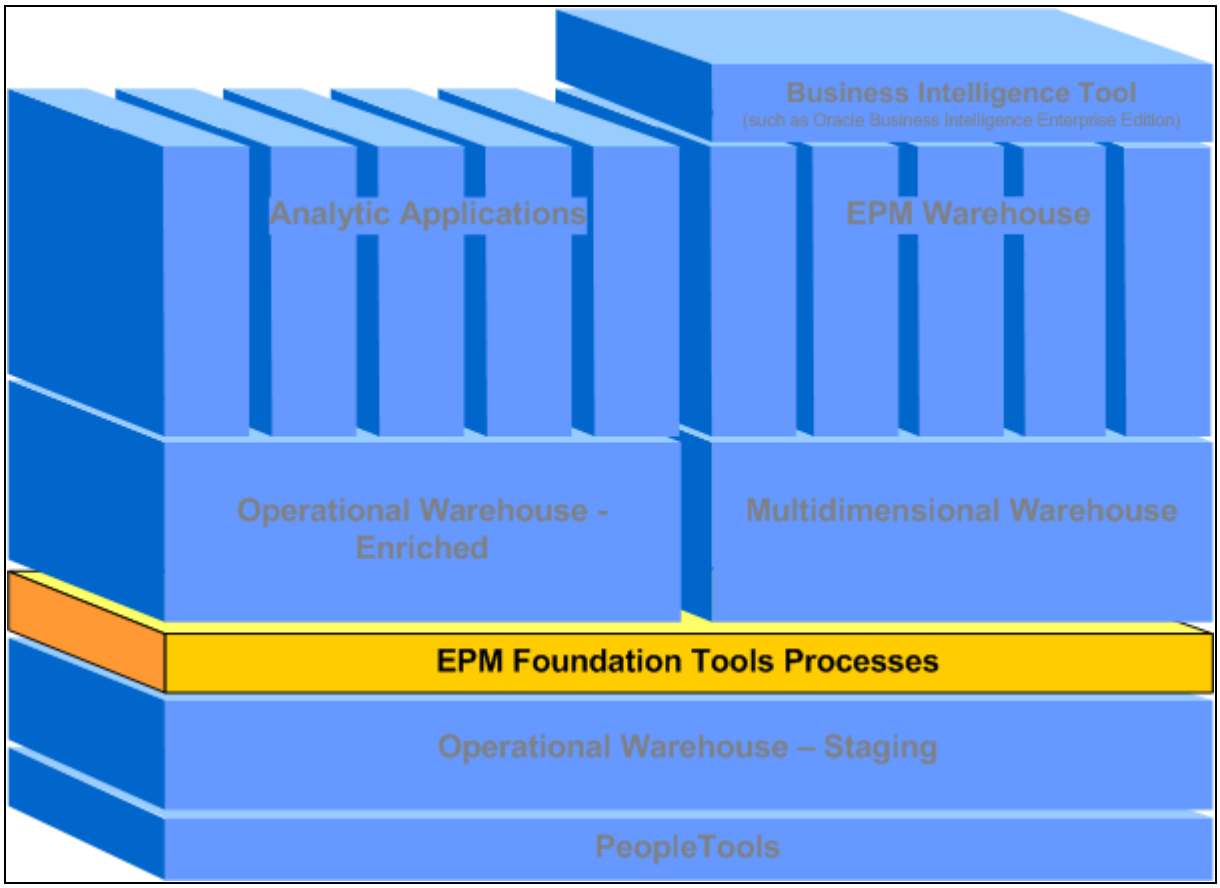
MDW Fact Table Naming Convention

MDW fact tables use the following naming convention: F_[*table name*].

EPM Foundation Toolset

EPM is delivered with EPM Foundation tools. These set of tools enable you to enrich, audit, and manage the rich content included with EPM with a high degree of automation. For example, the Clone Metadata tool enables you to quickly and easily create a duplicate copy of your existing metadata. EPM Foundation tools can be used with content included in the Operational Warehouse and the Multidimensional Warehouse.

The following graphic illustrates how the EPM foundation tools and processes fit into the overall EPM architecture.



EPM Foundation

The following sections provide additional details about EPM Foundation tools.

Setup Tools

Implementing EPM requires that you specify parameters within the warehouse that reflect your organization's basic business processes and parameters. For example, you must define parameters for unit of measure, country, and accounting calenders in EPM .

EPM delivers several setup tools which enable you to quickly and easily setup basic information in the warehouse including unit of measure, multiple language and currency, and operator defaults.

See Chapter 4, "Setting Up EPM Business Rules," page 45.

Security Tools

EPM security enables you to set up data access at a variety of entry points and control access to meet your business needs, right down to an individual field. Security tools enable you to:

- Use application security to control access to applications, menus, and objects. You can specify which applications are available to a group of users, which menus and EPM objects they can access.
- Use row-level security, for example, to implement dimension-level access to particular products, customers, or key performance metrics. This ensures that highly sensitive data is protected.

You can also set up a specific security for the IBM WebSphere ETL tool.

See Chapter 6, "Setting Up EPM Security," page 117.

Data Storage and Classification Tools

Implementing EPM involves configuring the system's structures to how your business operates. You can share common tables across reporting and analytical applications to minimize redundant data and system maintenance tasks.

Record metadata, for example, defines the first level of EPM metadata. It is used to identify and classify the tables that constitute the EPM data model. The record metadata identifies EPM tables as fact tables, fact reference tables, dimension tables, dimension reference tables, or transaction-dated tables. Each table is also classified to a specific data layer: the OWE or the MDW.

Tree manager provides an intuitive way to create, view, and maintain hierarchical definitions. An easy to understand user interface facilitates the creation and maintenance of trees. Tree mover enables you to moved PeopleSoft trees between different PeopleSoft application databases.

See Chapter 4, "Setting Up EPM Business Rules," page 45 and Chapter 16, "Setting Up and Working with Metadata for the Operational Warehouse - Enriched," page 339.

Performance Management Related Tools

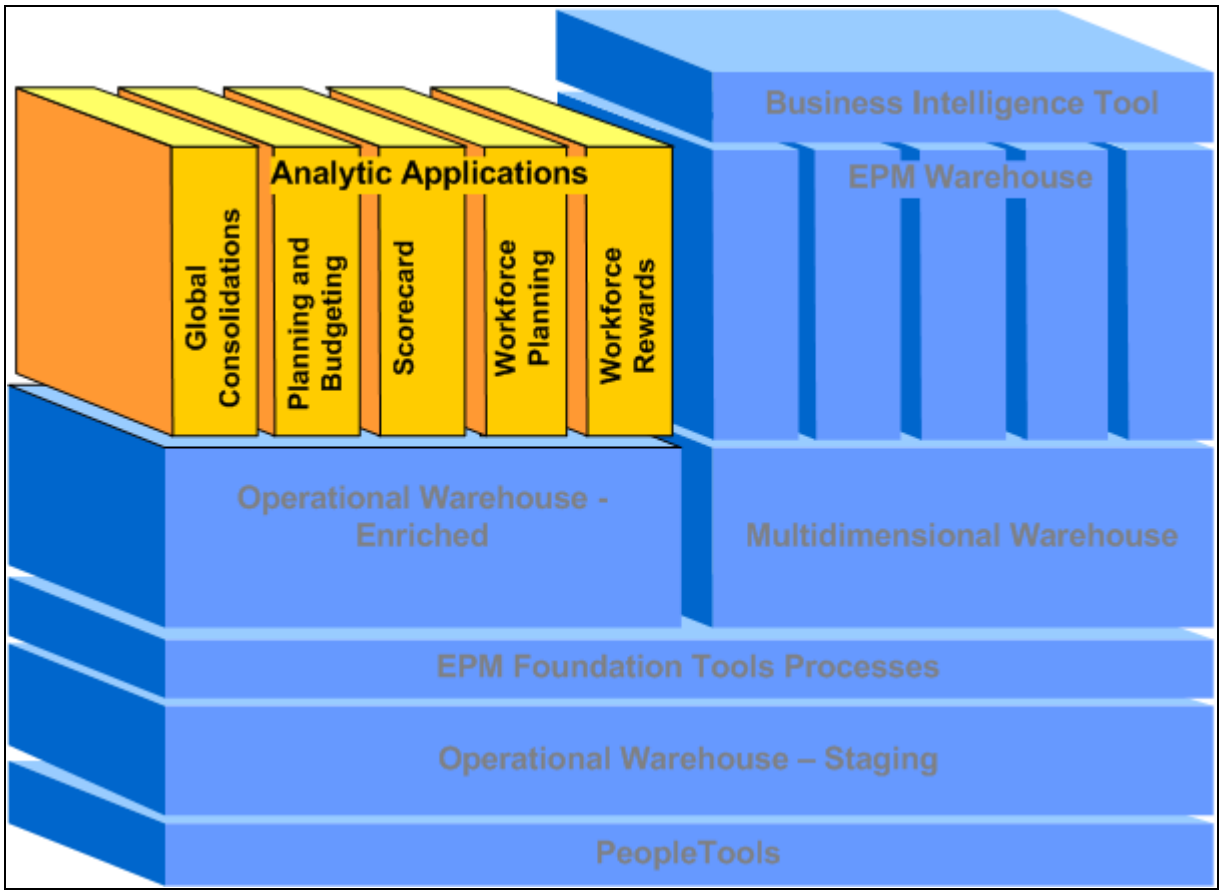
EPM utilizes shared components that provide functionality key to supporting high-volume analytical applications:

- Reusable filters and constraints stored in the metadata enable you to define sets of rules that can be shared across applications.
- Jobstreams streamline analytic processes and enable applications to run concurrently.

See Chapter 16, "Setting Up and Working with Metadata for the Operational Warehouse - Enriched," page 339 and Chapter 20, "Streamlining Processing with Jobstreams," page 461.

PeopleSoft EPM Analytical Applications

EPM provides the applications necessary to analyze business situations, model business scenarios, and monitor performance. The following graphic illustrates how the Analytical Applications fit into the overall EPM architecture.



EPM Analytical Applications

PeopleSoft Analytical Applications is comprised of the following individual applications:

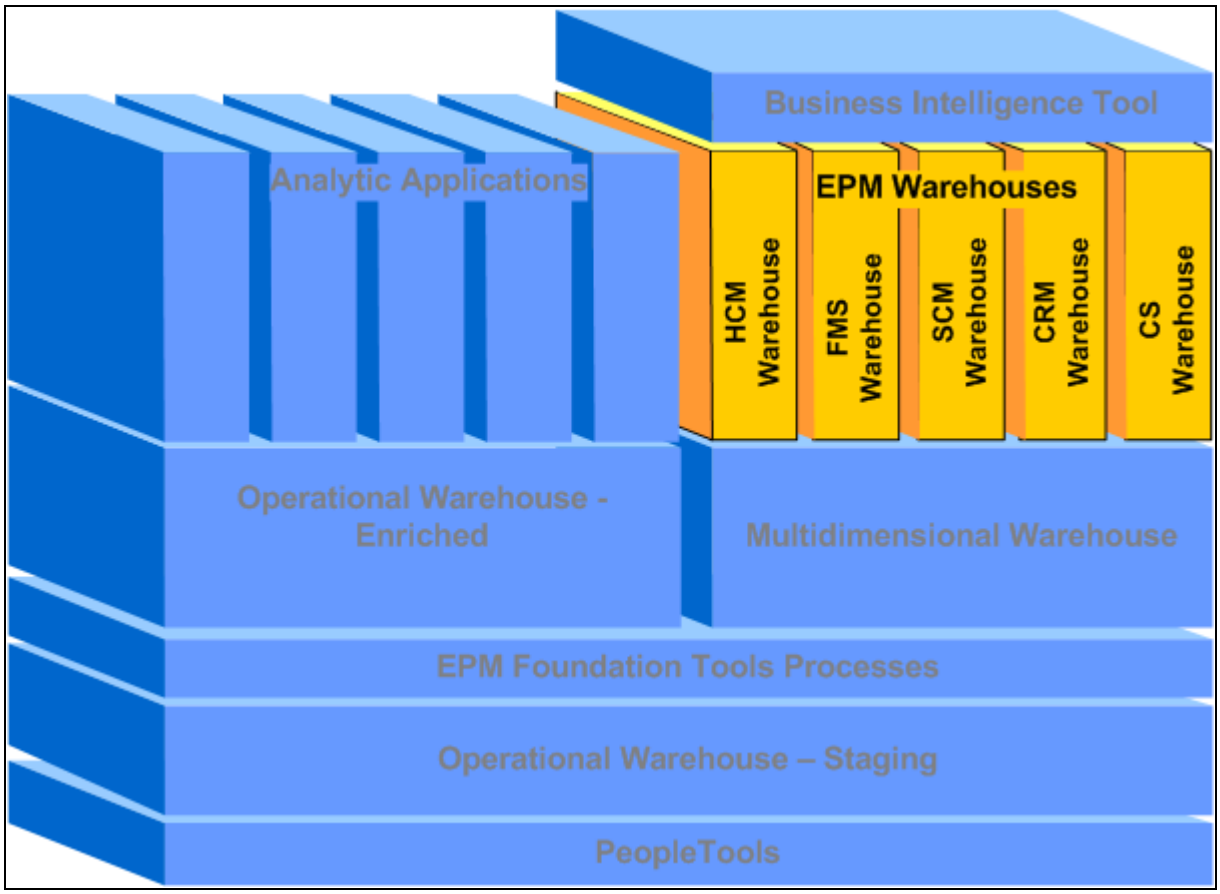
- PeopleSoft Enterprise Activity-Based Management
- PeopleSoft Enterprise Application Fundamentals for Financial Services Industry
- PeopleSoft Enterprise Funds Transfer Pricing
- PeopleSoft Enterprise Global Consolidations
- PeopleSoft Enterprise Performance Management Portal Pack
- PeopleSoft Enterprise Planning and Budgeting
- PeopleSoft Enterprise Project Portfolio Management
- PeopleSoft Enterprise Risk Weighted Capital
- PeopleSoft Enterprise Scorecard
- PeopleSoft Enterprise Workforce Analytics Applications
- PeopleSoft Enterprise Workforce Planning
- PeopleSoft Enterprise Workforce Rewards

For more details on the application or applications you have licensed, please refer to the specific PeopleBook or PeopleBooks.

PeopleSoft EPM Warehouses and Reporting

The PeopleSoft EPM warehouses provide you with the tools and technology to manage your organization's information that is used for reporting and analysis. Each warehouse is divided into multiple subject areas, or data marts. Each data mart is aligned with a business process, which enables you to answer strategic questions essential to your organization's bottom line.

The following graphic illustrates how the EPM warehouses fit into the overall EPM architecture.



EPM Warehouses

PeopleSoft provides the following EPM warehouses:

- Campus Solutions Warehouse
- Customer Relationship Management (CRM) Warehouse
- Financials Management Solution (FMS) Warehouse
- Financials Warehouse for Public Sector and Higher Education
- Human Capital Management (HCM) Warehouse

- Supply Chain Management (SCM) Warehouse

Prepackaged Content

PeopleSoft delivers the following content with each EPM warehouse:

- Extract Transform and Load (ETL) component
- Infrastructure tables and tools
- Security tables
- Staging tables
- Multidimensional Warehouse tables
- Data Models
- Measures

Reporting tables are built in the MDW to enable offloading of operational reports from your transactional systems. As part of your implementation, you need to consider which operational reports it makes sense to offload to the EPM warehouses.

The EPM warehouse PeopleBooks (such as the *PeopleSoft Enterprise Campus Solutions Warehouse PeopleBook*) provide more details on the each of the EPM warehouses.

Chapter 3

Implementing PeopleSoft Enterprise Performance Management

This chapter provides an overview of the tasks required to install, setup, and configure EPM and discusses how to use the EPM suite of PeopleBooks to locate implementation task instructions.

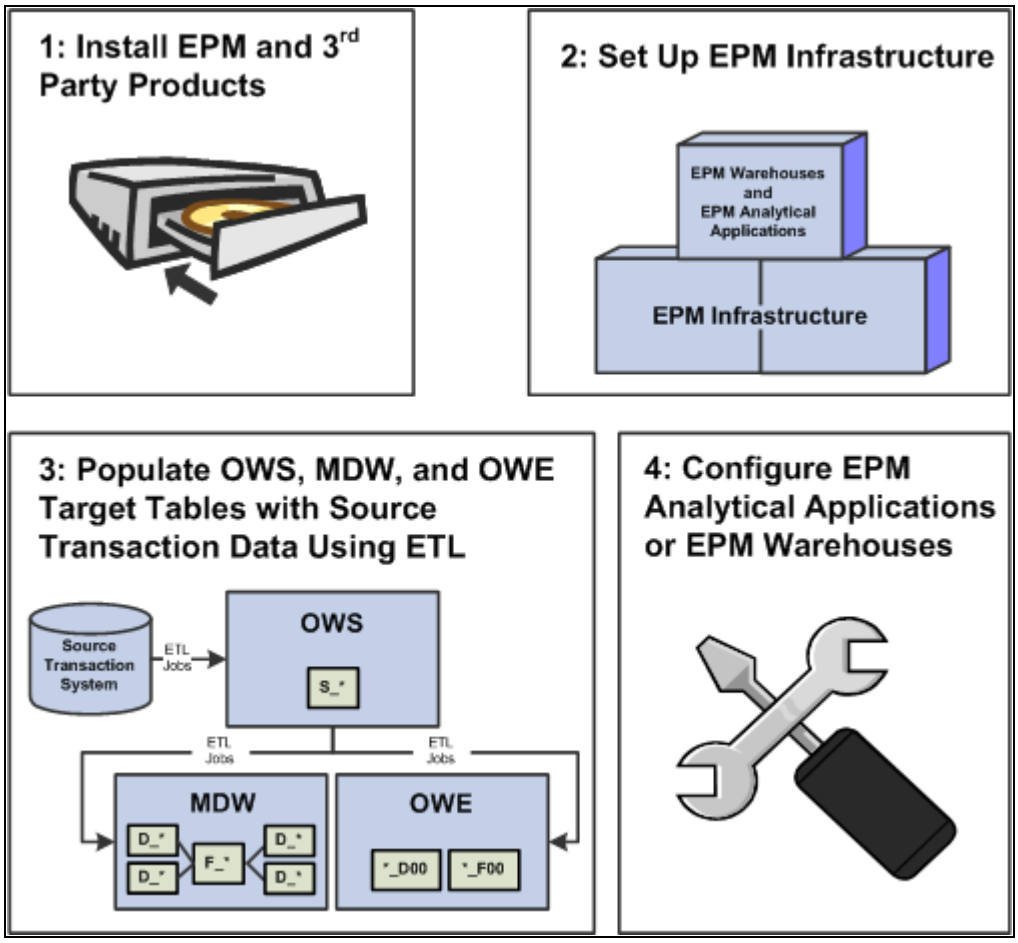
Understanding EPM Implementation

There is no singular approach to implementing EPM; each implementation varies depending on the EPM products you license. For example, many of the implementation tasks required to implement an EPM Analytical Application (such as Global Consolidations) are different from the implementation tasks required to implement an EPM Warehouse (such as the Campus Solutions Warehouse). However, there are some implementation tasks common to both EPM product lines and generally speaking, implementing EPM requires that you:

1. Install EPM and applicable third-party products (such as IBM WebSphere DataStage).
2. Set up EPM core infrastructure (which serves as the underlying framework for the EPM Warehouses and Analytical Applications).

   Examples of infrastructure setups include setting up currency codes (CURRENCY_CD_TB) and record metadata (META_REC_TBL).
3. Populate Operational Warehouse - Staging (OWS), Multidimensional Warehouse (MDW), and Operational Warehouse - Enriched (OWE) target warehouse tables with your source transaction data using the extract, transform, and load (ETL) tool.

   You also conform your disparate source transaction data using the ETL tool.
4. Configure either the EPM Warehouses or Analytical Applications for your business (depending on which EPM product you license).



Basic EPM Implementation Steps

The following sections will help you better understand which implementation tasks apply to your implementation and how to use the EPM suite of PeopleBooks to locate the instructions for those tasks.

Note. PeopleSoft Setup Manager can also help you determine which specific tasks are required for your implementation by generating a list of setup tasks based on the features you license. The list of setup tasks include the components that you must set up, listed in the order in which you must enter data into the component tables, and links to the corresponding PeopleBook documentation.

EPM Installation Tasks

The first step in implementing EPM is to install all the necessary software on your designated machine(s).

The following table provides an *example* of some of the installation tasks you perform to implement EPM:

Task	***Common, EPM Warehouses, or Analytical Applications?***	***Documentation Resource***	***Documentation Location***
Pre-installation	Common	PeopleSoft Pre-Installation Checklist	See My Oracle Support.

Task	Common, EPM Warehouses, or Analytical Applications?	Documentation Resource	Documentation Location
Review hardware/software requirements	Common	PeopleSoft Hardware and Software Guide	See My Oracle Support.
Install IBM WebSphere DataStage	Common	*PeopleSoft Enterprise Performance Management Installation Guide* *IBM Information Server: Planning Installation and Configuration Guide*	Installation CD
Install EPM	Common	*PeopleSoft Enterprise Performance Management Installation Guide*	Installation CD

EPM Core Infrastructure and ETL Setup Tasks

The second step in implementing EPM is to set up EPM infrastructure tables and populating warehouse target tables with source transaction data. Some of these tasks include defining currency conversion methodology, setting up warehouse business units, and configuring ETL environmental parameters.

The core infrastructure and ETL setup tasks are grouped as follows:

- Core infrastructure setups.
- Common ETL setups.

Core Infrastructure Setups

The following table provides an *example* of some of the core EPM infrastructure setup tasks you perform to implement EPM:

Task	Common, EPM Warehouses, or Analytical Applications?	Documentation Resource	Documentation Location
Specify EPM Sources	Common	*Setting Up EPM Business Rules* PeopleSoft Enterprise Performance Management Fundamentals PeopleBook	See Chapter 4, "Setting Up EPM Business Rules," Specifying Your EPM Sources, page 50.
Specify Country and State Info	Common	*Setting Up EPM Business Rules* PeopleSoft Enterprise Performance Management Fundamentals PeopleBook	See Chapter 4, "Setting Up EPM Business Rules," Setting Up Country and State Information, page 53.

Task	*Common, EPM Warehouses, or Analytical Applications?*	*Documentation Resource*	*Documentation Location*
Define Accounting Calendars	Common	*Setting Up EPM Business Rules* PeopleSoft Enterprise Performance Management Fundamentals PeopleBook	See Chapter 4, "Setting Up EPM Business Rules," Defining Accounting Calendars, page 59.
Set Up Gregorian Calendar	EPM Warehouses	*Setting Up EPM Business Rules* PeopleSoft Enterprise Performance Management Fundamentals PeopleBook	See Chapter 4, "Setting Up EPM Business Rules," Setting Up the Gregorian Calendar [EPM Warehouses], page 73.
Define Units of Measure	Common	*Setting Up EPM Business Rules* PeopleSoft Enterprise Performance Management Fundamentals PeopleBook	See Chapter 4, "Setting Up EPM Business Rules," Defining Units of Measure, page 76.
Define Dimensions	Common	*Setting Up EPM Business Rules* PeopleSoft Enterprise Performance Management Fundamentals PeopleBook	See Chapter 4, "Setting Up EPM Business Rules," Defining and Maintaining Dimensions, page 77.
Set Up Currency Tables	Common	*Setting Up Currency Rules for EPM* PeopleSoft Enterprise Performance Management Fundamentals PeopleBook	See Chapter 5, "Setting Up Currency Rules for EPM," Setting Up EPM Currency Tables, page 95.
Define Market Rates	Common	*Setting Up Currency Rules for EPM* PeopleSoft Enterprise Performance Management Fundamentals PeopleBook	See Chapter 5, "Setting Up Currency Rules for EPM," Setting Up Market Rates for EPM Currency Conversion, page 97.
Define Currency Quotations	Common	*Setting Up Currency Rules for EPM* PeopleSoft Enterprise Performance Management Fundamentals PeopleBook	See Chapter 5, "Setting Up Currency Rules for EPM," Defining Currency Quotations for EPM Currency Conversion, page 101.
Set Up Currency Rate Calculations	Common	*Setting Up Currency Rules for EPM* PeopleSoft Enterprise Performance Management Fundamentals PeopleBook	See Chapter 5, "Setting Up Currency Rules for EPM," Calculating Currency Rates for EPM Currency Conversion, page 111.

Task	Common, EPM Warehouses, or Analytical Applications?	Documentation Resource	Documentation Location
Set Up EPM Security	Common	*Setting Up EPM Security* PeopleSoft Enterprise Performance Management Fundamentals PeopleBook	See Chapter 6, "Setting Up EPM Security," page 117.

Common ETL Setups

The following table provides an *example* of some of the common ETL setup tasks you perform to implement EPM:

Task	Common, EPM Warehouses, or Analytical Applications?	Documentation Resource	Documentation Location
Configure IBM WebSphere DataStage for EPM	Common	*Preparing to Load Source Data Into EPM* *Setting Up DataStage for EPM* PeopleSoft Enterprise Performance Management Fundamentals PeopleBook	See Chapter 7, "Preparing to Load Source Data Into EPM," page 161 and Chapter 11, "Setting Up DataStage for EPM," page 255.
Specify ETL parameters	Common	*Defining ETL Parameters* PeopleSoft Enterprise Performance Management Fundamentals PeopleBook	See Chapter 12, "Defining ETL Parameters," page 265.
Run initial setup jobs	Common	*Running Initial Setup Jobs* PeopleSoft Enterprise Performance Management Fundamentals PeopleBook	See Chapter 13, "Running Initial Setup Jobs," page 275.
Import source business units into EPM	Common	*Importing Source Business Units into EPM to Create Warehouse Business Units* PeopleSoft Enterprise Performance Management Fundamentals PeopleBook	See Chapter 14, "Importing Source Business Units into EPM to Create Warehouse Business Units," page 283.

OWE / MDW Specific Setup Tasks

You may recall from the chapter, *Understanding PeopleSoft Enterprise Performance Management,* that the OWE structure stores data arranged in a normalized format for the Analytical Applications, and the MDW structure stores data arranged in a dimensional schema for the EPM Warehouses. Hence, implementation tasks related to the MDW affect the EPM Warehouses and tasks related to the OWE affect the Analytical Applications.

After all installation and core EPM implementation tasks are completed, the final step is to perform specific implementation tasks that apply only to either the EPM Warehouses or the Analytical Applications. For example, if you purchased the Global Consolidations analytical application you must set up record metadata, which is an implementation task specific to the analytical applications.

MDW Specific Setups

MDW specific setups are documented in EPM warehouse PeopleBooks (for example, *PeopleSoft Enterprise Campus Solutions Warehouse PeopleBook*).

The following table provides an *example* of some of the MDW specific setup tasks you perform to implement an EPM warehouse:

Task	***Documentation Resource***
Run Global Dimension jobs for your specific warehouse (Campus Solutions Warehouse, for example).	*Running [product name] Warehouse Implementation Jobs* (for example, Running Campus Solutions Warehouse Implementation Jobs) PeopleSoft Enterprise [product name] Warehouse PeopleBook (for example, *PeopleSoft Enterprise Campus Solutions Warehouse PeopleBook)*
Run SKU jobs for your specific warehouse (Supply Chain Management Warehouse, for example)	*Running [product name] Warehouse Implementation Jobs* PeopleSoft Enterprise [product name] Warehouse PeopleBook
Configure slowly changing dimensions	*Configuring Slowly Changing Dimensions* PeopleSoft Enterprise [product name] Warehouse PeopleBook
Implement currency conversion	*Implementing Currency Conversion* PeopleSoft Enterprise [product name] Warehouse PeopleBook
Implement multilanguage conversion	*Setting Up Multilanguage Processing and Running the Language Swap Utility* PeopleSoft Enterprise [product name] Warehouse PeopleBook
Denormalize trees and recursive hierarchies	*Processing Trees and Recursive Hierarchies* PeopleSoft Enterprise [product name] Warehouse PeopleBook

PeopleSoft EPM provides the following EPM warehouse PeopleBooks, where you can find MDW specific setup tasks to implement an EPM warehouse:

- PeopleSoft Enterprise Campus Solutions Warehouse PeopleBook
- PeopleSoft Enterprise Customer Relationship Management Warehouse PeopleBook
- PeopleSoft Enterprise Financial Management Solution Warehouse PeopleBook

- Financials Warehouse for Public Sector and Higher Education PeopleBook
- PeopleSoft Enterprise Human Capital Management Warehouse PeopleBook
- PeopleSoft Enterprise Supply Chain Management Warehouse PeopleBook

OWE Setups

OWE specific setups are documented in both this PeopleBook and EPM Analytical Application PeopleBooks (for example, PeopleSoft Enterprise Global Consolidations PeopleBook).

The following table provides an *example* of some of the OWE specific setup tasks you perform to implement an EPM Analytical Application:

Task	*Documentation Resource*	*Documentation Location*
Define record metadata	*Setting Up and Working with Metadata for the Operational Warehouse - Enriched* PeopleSoft Enterprise Performance Management Fundamentals PeopleBook	See Chapter 16, "Setting Up and Working with Metadata for the Operational Warehouse - Enriched," Setting Up Record Metadata, page 348.
Set up datamaps	*Setting Up and Working with Metadata for the Operational Warehouse - Enriched* PeopleSoft Enterprise Performance Management Fundamentals PeopleBook	See Chapter 16, "Setting Up and Working with Metadata for the Operational Warehouse - Enriched," Setting Up Datamaps, page 363.
Specify ledger mapping defaults	*Setting Up Business Rules for the Operational Warehouse - Enriched* PeopleSoft Enterprise Performance Management Fundamentals PeopleBook	See Chapter 18, "Setting Up Business Rules for the Operational Warehouse - Enriched," Specifying Ledger Mapping Defaults, page 425.
Implement currency conversion	*Setting Up Business Rules for the Operational Warehouse - Enriched* PeopleSoft Enterprise Performance Management Fundamentals PeopleBook	See Chapter 18, "Setting Up Business Rules for the Operational Warehouse - Enriched," Setting Up and Running Currency Conversion, page 443.
Define jobstreams	*Streamlining Processing with Jobstreams* PeopleSoft Enterprise Performance Management Fundamentals PeopleBook	See Chapter 20, "Streamlining Processing with Jobstreams," page 461.
Define allocation rules	*Using Data Enrichment Tools* PeopleSoft Enterprise Performance Management Fundamentals PeopleBook	See Chapter 22, "Using Data Enrichment Tools," Defining Allocation Manager Rules, page 572.

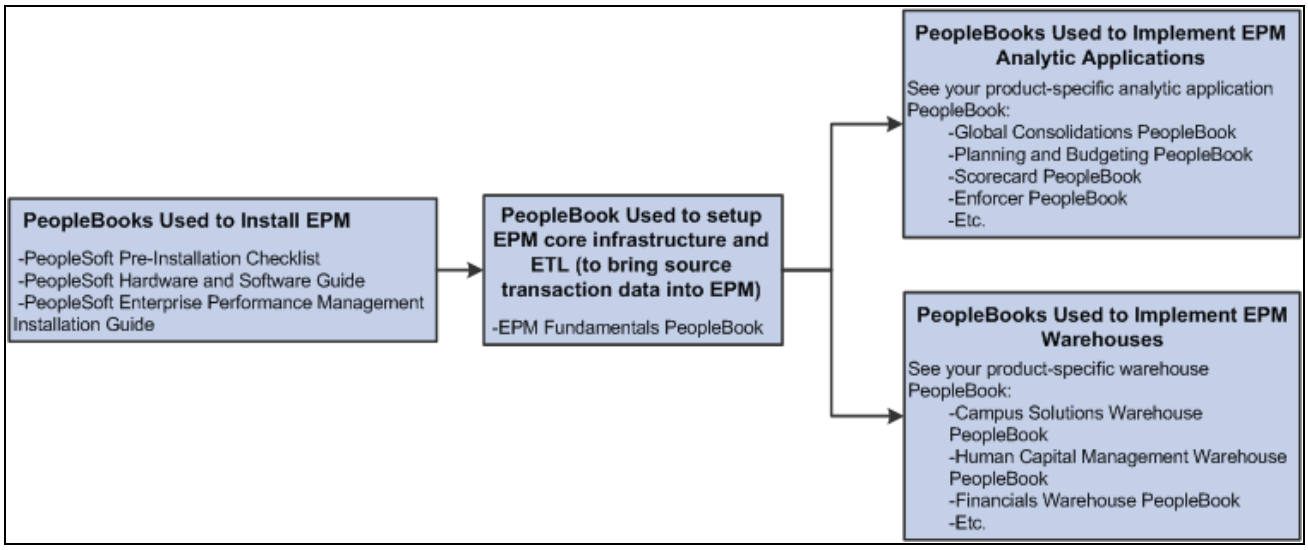
Task	*Documentation Resource*	*Documentation Location*
Define your organizational structure (example of a setup specifically for the Global Consolidations analytical application)	*Defining Your Organizational Structure* PeopleSoft Enterprise Global Consolidations PeopleBook	See *PeopleSoft Enterprise Global Consolidations 9.1 PeopleBook*, "Defining Your Organizational Structure."
Establish KPIs (example of a setup specifically for the Scorecard analytical application)	*Establishing and Maintaining KPIs* PeopleSoft Enterprise Scorecard PeopleBook	See *PeopleSoft Enterprise Scorecard 9.1 PeopleBook*, "Establishing and Maintaining KPIs."

Using the EPM Suite of PeopleBooks to Locate Implementation Task Instructions

The suite of EPM PeopleBooks are organized into four general categories:

- EPM Installation PeopleBooks: These books contain EPM installation information, hardware and software guidelines, and third-party product installation information (such as IBM WebSphere DataStage).
- EPM Fundamentals PeopleBook: This book contains core infrastructure setup common to all EPM products, ETL setup and processing information, and some setups for the EPM Warehouses and Analytical Applications.
- EPM Warehouses PeopleBooks: These books contain specific set up and configuration information particular to each EPM warehouse.
- EPM Analytical Applications PeopleBooks: These books contain specific setup and configuration information particular to each analytical application.

The organization of the suite of EPM PeopleBooks represent the various steps required in an EPM implementation (installation, core infrastructure setup, ETL setup and processing, and EPM Warehouse or Analytical Application configuration). You progress through the different EPM PeopleBooks in the same order you progress through an implementation, using the installation books first and the warehouse or analytical application books last:



EPM PeopleBook Usage Order

The following table lists all available EPM documentation by category:

EPM Installation PeopleBooks	***EPM Fundamentals PeopleBook***	***EPM Warehouse PeopleBooks***	***EPM Analytical Application PeopleBooks***
PeopleSoft Pre-Installation Checklist	*PeopleSoft Enterprise Performance Management Fundamentals PeopleBook*	*PeopleSoft Enterprise Campus Solutions Warehouse PeopleBook*	*PeopleSoft Enterprise Activity-Based Management PeopleBook*
PeopleSoft Enterprise Performance Management Hardware and Software Requirements Guide		*PeopleSoft Enterprise Customer Relationship Management Warehouse PeopleBook*	*PeopleSoft Enterprise Application Fundamentals for Financial Services Industry PeopleBook*
PeopleSoft Enterprise Performance Management Installation Guide		*PeopleSoft Enterprise Financial Solutions Management Warehouse PeopleBook*	*PeopleSoft Enterprise Funds Transfer Pricing PeopleBook*
		Financials Warehouse for Public Sector and Higher Education PeopleBook	*PeopleSoft Enterprise Global Consolidations PeopleBook*
		PeopleSoft Enterprise Human Capital Management Warehouse PeopleBook	*PeopleSoft Enterprise Performance Management Portal Pack PeopleBook*
		PeopleSoft Enterprise Supply Chain Management Warehouse PeopleBook	*PeopleSoft Enterprise Planning and Budgeting PeopleBook*
			PeopleSoft Enterprise Project Portfolio Management PeopleBook

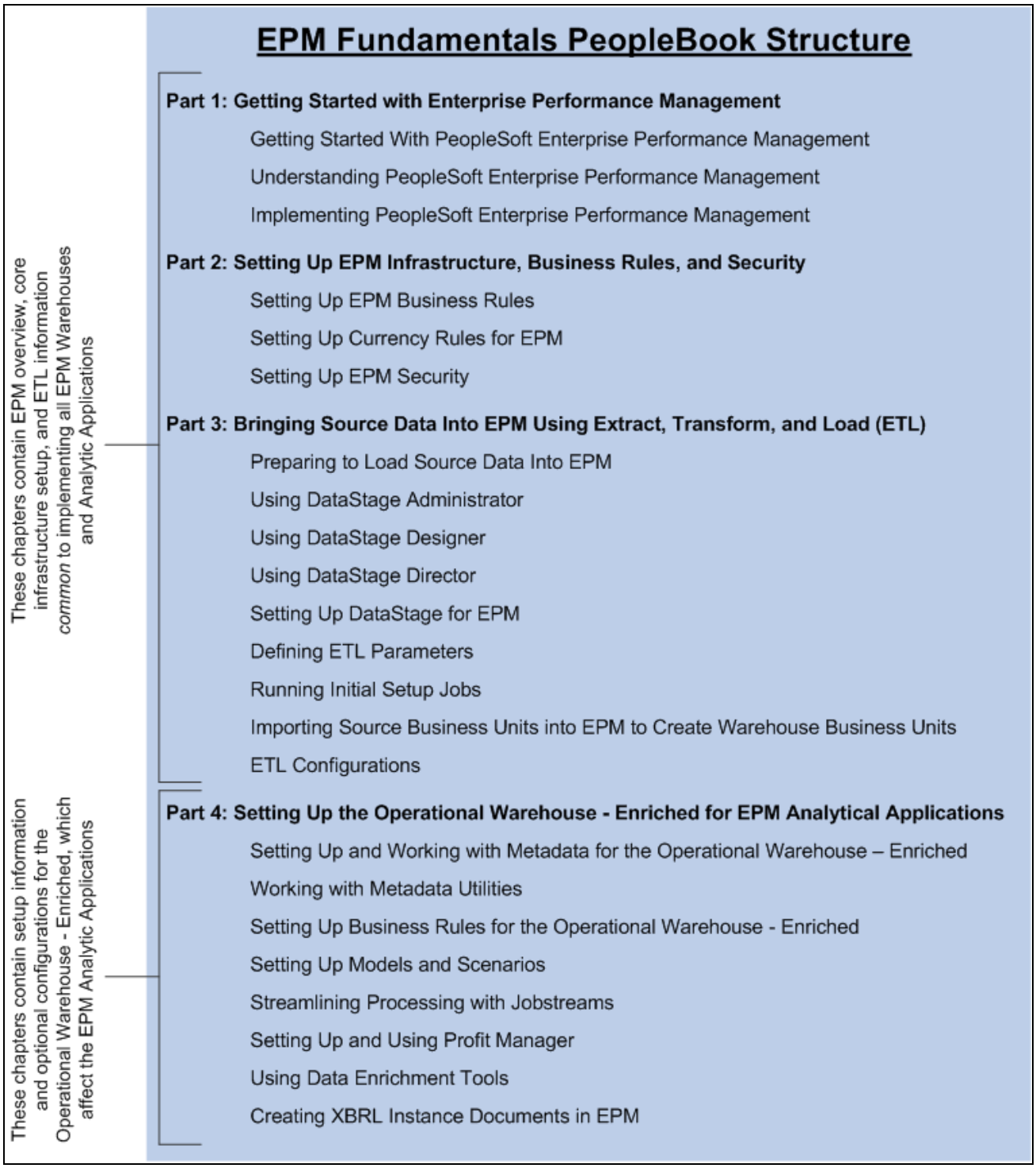
EPM Installation PeopleBooks	*EPM Fundamentals PeopleBook*	*EPM Warehouse PeopleBooks*	*EPM Analytical Application PeopleBooks*
			PeopleSoft Enterprise Risk Weighted Capital PeopleBook
			PeopleSoft Enterprise Scorecard PeopleBook
			PeopleSoft Enterprise Workforce Analytic Applications PeopleBook
			PeopleSoft Enterprise Workforce Planning PeopleBook
			PeopleSoft Enterprise Workforce Rewards PeopleBook

Note. If you are implementing EPM with the assistance of a PeopleSoft consultant, the consultant can access a searchable, online version of the aforementioned PeopleBooks from My Oracle Support.

Using This PeopleBook to Locate Implementation Tasks

The structure of the *PeopleSoft Enterprise Performance Management Fundamentals PeopleBook* is designed to help you locate the specific tasks required for your implementation, which depends on the EPM product you license. The structure of this PeopleBook separates chapters and implementation tasks according to whether they relate to all EPM products (common), or only the OWE and the EPM Analytical Applications. Implementation tasks for the MDW and EPM Warehouses are located in separate PeopleBooks.

Chapters located in parts 1, 2, and 3 of this PeopleBook contain an overview of EPM and core infrastructure and ETL setup information necessary to implement all EPM products (warehouses and analytical applications). Chapters located in part 4 of this PeopleBook contain setup information and optional configurations for the OWE, which are necessary for implementing the EPM Analytical Applications.



Structure of the EPM Fundamentals PeopleBook

Regardless of the EPM product you purchase, you must perform the tasks documented in the core infrastructure and ETL setup chapters (see preceding graphic), as these tasks are required for all EPM products. However, if you purchase an EPM Analytical Application, you must also perform the tasks documented in OWE-related chapters (see preceding graphic). If you purchase an EPM Warehouse, you need not perform the tasks documented in the OWE-related chapters, but you must refer to EPM Warehouse implementation tasks in your warehouse-specific PeopleBook (for example, *PeopleSoft Enterprise Campus Solutions Warehouse PeopleBook*).

Part 2

Setting Up EPM Infrastructure, Business Rules, and Security

Chapter 4

Setting Up EPM Business Rules

This chapter provides overviews of PeopleSoft EPM business rule setups and discusses how to:

- Review installed products.
- Specify your EPM sources.
- Set up country and state information.
- Define accounting calendars.
- Set up the Gregorian calendar. [EPM Warehouses]
- Set up time zones. [EPM Warehouses]
- Define units of measure.
- Define and maintain dimensions.
- Define operator defaults.
- Archive EPM data.
- Set up chart viewing and printing results [Analytical Applications]
- Set up ledger-based nVision reporting for EPM

Note. When the term *EPM Warehouses* or *Analytical Applications* appears in the task titles below, it means that the task applies only to that particular EPM product line. When neither term appears in the task title, it means the task applies to both EPM product lines.

Understanding PeopleSoft EPM Business Rule Setups

After installing EPM, you must set up the infrastructure, key business rules, and processing variables that support the product. These rules provide the foundation on which EPM operates and can include country and state, unit of measure, and calendar rules. Because these rules act as a foundation and span the entire EPM product suite, you must perform these setup tasks prior to beginning other EPM implementation tasks documented in this guide.

The business rules and processing variables that you define for EPM are stored in EPM database tables. You define these rules and populate the EPM tables using the PeopleSoft Pure Internet Architecture (PIA) pages in the EPM Foundation setup menu. These pages are built over EPM tables that are shared across many EPM products.

Note. Please be aware that the order in which the setup tasks are presented in this chapter do not necessarily indicate the actual order in which they should be performed during implementation. It is highly recommended that you use Setup Manager to determine the implementation tasks that are required for your organization and the related implementation sequence.

Reviewing Installed Products

Before you can begin working with any PeopleSoft EPM application, you must specify installation options to indicate how you plan to use the system and which applications you will be using. Installation options are defined for your entire database—they are not specific to a business unit or setID.

This section discusses how to:

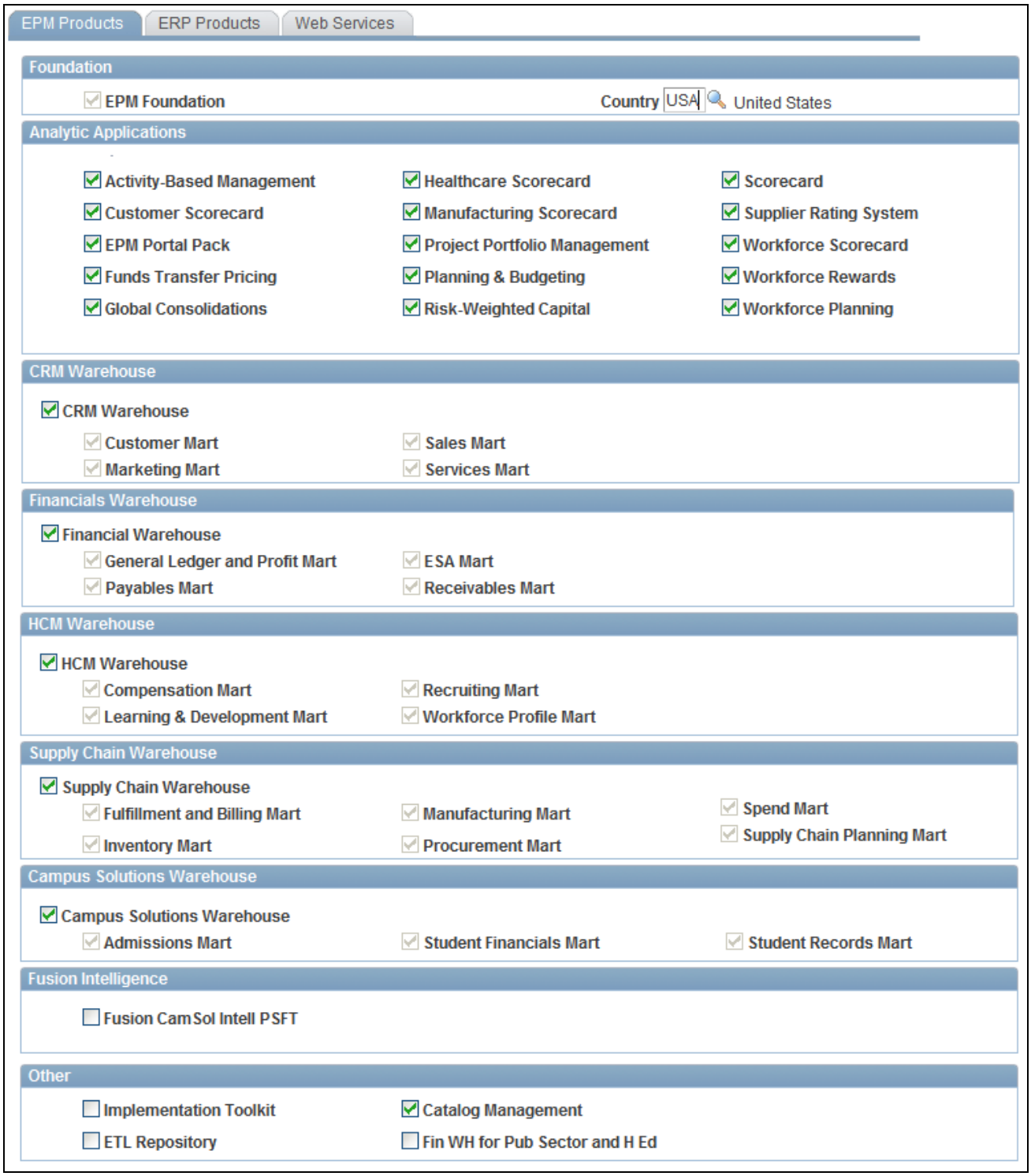
- Review installed PeopleSoft EPM products.
- Review installed PeopleSoft Enterprise Resource Planning (ERP) products.
- Set Web Services options.

Pages Used to Set Installation Options

Page Name	*Definition Name*	*Navigation*	*Usage*
EPM Products	INSTALLATION_PF1	EPM Foundation, EPM Setup, Installation Analysis & Options, Installation Options, EPM Products	Review installed PeopleSoft Enterprise Performance Management products.
ERP Products	INSTALLATION_PF2	EPM Foundation, EPM Setup, Installation Analysis & Options, Installation Options, ERP Products	Review installed PeopleSoft ERP products.
Web Services	INSTALLATION_PF3	EPM Foundation, EPM Setup, Installation Analysis & Options, Installation Options, Web Services	Set web services options.

Reviewing Installed PeopleSoft EPM Products

Access the EPM Products page (EPM Foundation, EPM Setup, Installation Analysis & Options, Installation Options, EPM Products).



EPM Products page

Foundation

If you have installed the EPM Foundation toolset on your database, it is automatically selected here.

Use the Country field to select a default country, which determines the address format for pages that use address fields.

You define the address formats on the Country Table - Address Format page.

Analytic Applications

Any EPM analytical applications that have been installed in your database are automatically selected.

CRM Warehouse

If you have installed the CRM Warehouse on your database, the warehouse and its data marts are automatically selected here.

Financials Warehouse

If you have installed the CRM Warehouse on your database, the warehouse and its data marts are automatically selected here.

HCM Warehouse

If you have installed the HCM Warehouse on your database, the warehouse and its data marts are automatically selected here.

Supply Chain Warehouse

If you have installed the SCM Warehouse on your database, the warehouse and its data marts are automatically selected here.

Campus Solutions Warehouse

If you have installed the Campus Solutions Warehouse on your database, the warehouse and its data marts are automatically selected here.

Fusion Intelligence

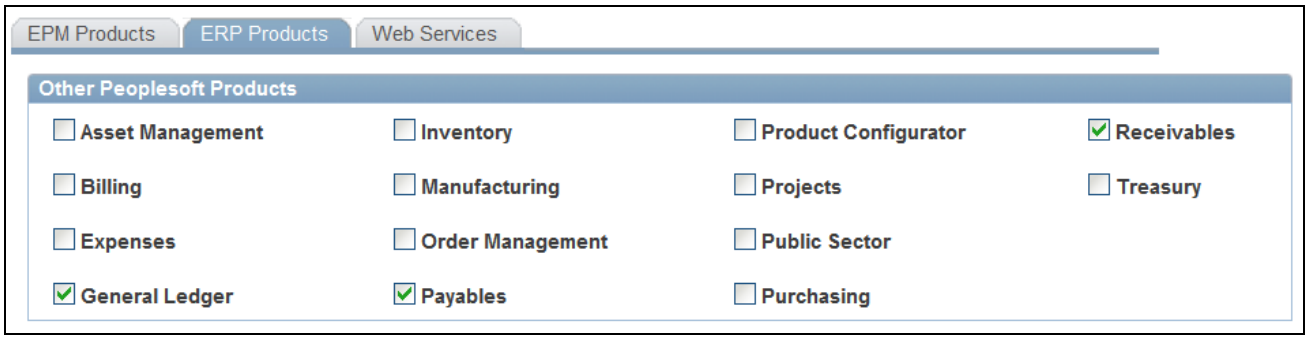
If you have installed the Fusion Campus Solutions Intelligence application on your database, it is automatically selected here.

Other

Any other products that relate to your EPM implementation are automatically selected.

Reviewing Installed PeopleSoft ERP Products

Access the ERP Products page (EPM Foundation, EPM Setup, Installation Analysis & Options, Installation Options, ERP Products).



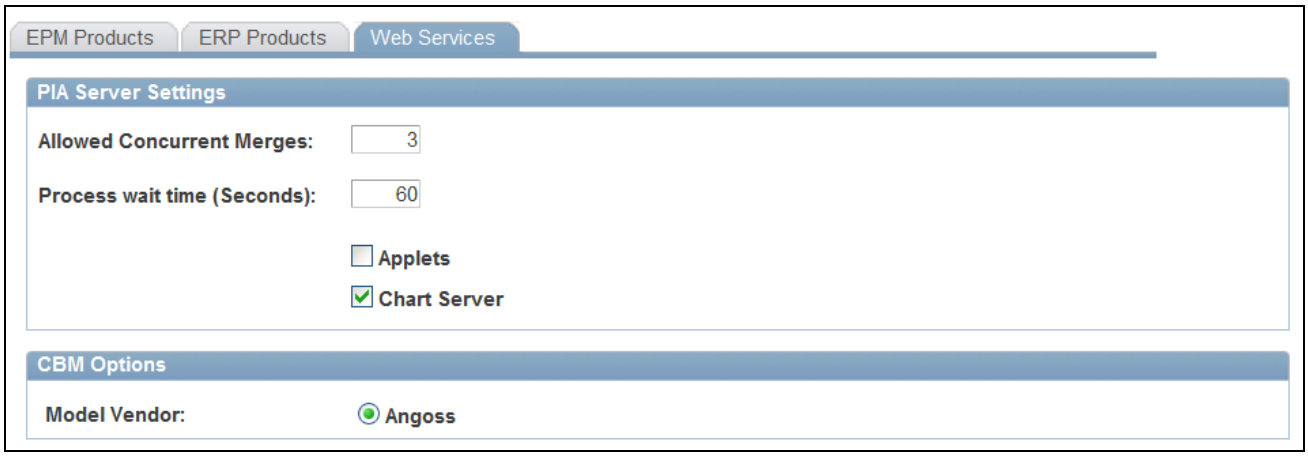
ERP Products page

Other PeopleSoft Products

Any other PeopleSoft ERP products that have been installed in your database are automatically selected.

Setting Web Services Options

Access the Web Services page (EPM Foundation, EPM Setup, Installation Analysis & Options, Installation Options, Web Services).



Web Services page

Allowed Concurrent Merges Enter the number of concurrent merges allowed in your installation. The default is *3*, which matches the number of record suites that are delivered with your system.

Process wait time (Seconds) Enter the process wait time. The default is *60* seconds. Waiting improves performance of processes by eliminating repeated database requests.

Warning! Because each merge job in a jobstream reads in the number of concurrent updates, changing its value while merge processes are running can have unexpected results. Changes to the process wait time affect only subsequent runs of the engine.

Applets Select if you are using applets with PeopleSoft Activity-Based Management or Enterprise Scorecard. See the application-specific PeopleBooks for more information on using the chart server and enabling applets.

Chart Server Select if you are using the chart server to create charts through PeopleSoft Activity-Based Management, Enterprise Scorecard, or the financial services industry applications (PeopleSoft Risk-Weighted Capital, Asset Liability Management, or Funds Transfer Pricing).

See Also

Chapter 20, "Streamlining Processing with Jobstreams," page 461

PeopleSoft Enterprise Scorecard 9.1 PeopleBook, "Monitoring Scorecards and KPIs," Viewing Scorecard Results

Specifying Your EPM Sources

This section provides an overview of the relationship between source system data and EPM, and discusses how to define warehouse sources.

Understanding Source System Data and EPM

EPM enables you to extract, transform, and consolidate data from multiple source transaction systems into a series of target warehouse tables in the EPM database. This means that the EPM database is able to accommodate data from several Enterprise and third-party sources. Although the data is commingled in the EPM database, certain EPM processes require the ability to trace the data in target warehouse tables to its original source (original source transaction system). Other EPM processes use specific attributes associated with the source transaction system to process data. For example, the *language swap utility* uses the source system's base language for multiple language processing.

Because source system information is required, you must define the source in EPM and specify the attributes associated with that source using the Define Warehouse Sources page. Information specified on this page include source system ID (SRC_SYS_ID), base currency, and base language, and is used in several different EPM processes. Most notably, the source system ID is used in the ETL process and helps to maintain source data history as well as data uniqueness. Currency and rate type are used in multiple currency processing. The source system ID and default setID are used later to define warehouse business units.

Single Signon and Your GL Source Database

The Global Consolidations analytical application has a feature that enables you to drill down to your source GL database from the consolidation audit using the PeopleTools *single signon* feature. The single signon feature enables you to access additional PeopleSoft Enterprise FSCM transaction databases without entering an ID or a password after you have already been authenticated in another PeopleSoft database.

In order to use the Global Consolidations GL drill-down feature you must set up the single signon feature for each GL source in PeopleTools. You must also use the Define Warehouse Sources page to associate the GL source system ID with its corresponding single signon portal information. This information is used to build the URL to transfer to a GL instance.

See *PeopleSoft Enterprise Global Consolidations 9.1 PeopleBook*, "PeopleSoft Enterprise Global Consolidations Preface."

See *PeopleSoft Enterprise PeopleTools PeopleBook: Security Administration*

See Also

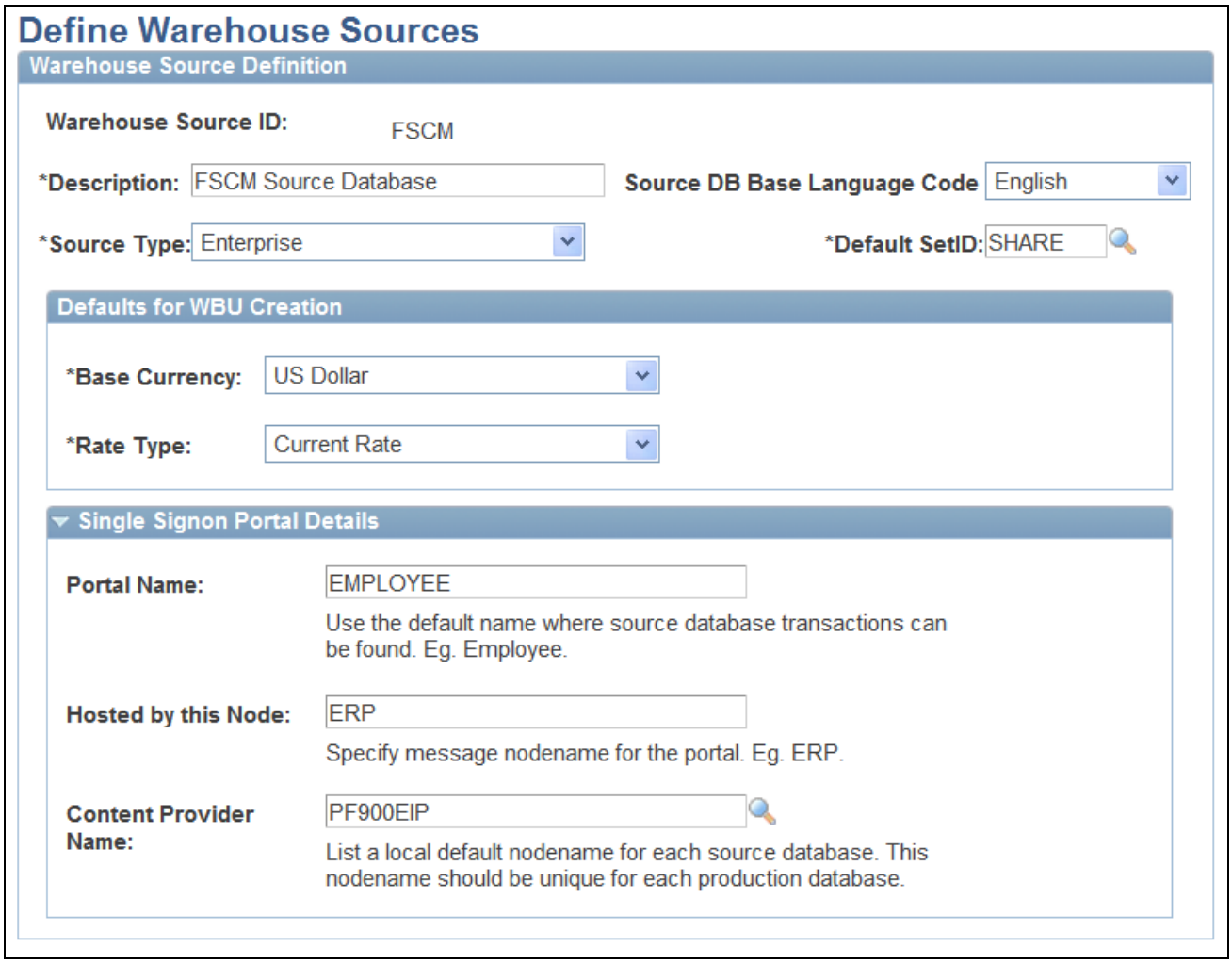
Chapter 14, "Importing Source Business Units into EPM to Create Warehouse Business Units," page 283

Page Used to Specify EPM Sources

Page Name	*Definition Name*	*Navigation*	*Usage*
Define Warehouse Sources	PF_WAREHOUSE_SRC	EPM Foundation, EPM Setup, Warehouse Sources & Bus Units, Define Warehouse Sources	Specify a source ID for a source transaction system and define its basic attributes, such as base language and currency.

Defining Warehouse Sources

Access the Define Warehouse Sources page (EPM Foundation, EPM Setup, Warehouse Sources & Bus Units, Define Warehouse Sources).



Define Warehouse Sources page

Warehouse Source ID Displays the unique source ID for the source transaction system from which you are extracting data.

This code can be up to five characters long, unique, and is defined when you add a new source system. The source ID is primarily used for ETL processing.

Source DB Base Language Code (source database base language code) Select the base language used by the specified source transaction system.

This code is used for multilanguage processing for the EPM warehouses (for example, the Campus Solutions Warehouse).

Source Type Specify whether the source transaction system is an *Enterprise* or *Other* source.

Default Set ID Enter a default setID for the tableset associated with the selected source system.

Defaults for WBU Creation (Defaults for Warehouse Business Unit Creation)

Base Currency Select the base currency used by the selected source system.

Base currency is used in multiple currency processing and the creation of warehouse business units.

Rate Type Select the rate type used by the selected source system.

Rate type is used in multiple currency processing and the creation of warehouse business units.

Single Signon Portal Details (Optional)

Use this section to associate a GL source system ID with its corresponding single signon portal information. Completing this section is optional if you are not implementing the GL drill down feature of the Global Consolidations analytical application.

Portal Name Enter the single signon portal name where the source database transaction data can be found.

Hosted by this Node Enter the name of the node hosting the portal.

Content Provider Name Enter the name of the local default node for the source database.

Setting Up Country and State Information

This section discusses how to:

- Review country descriptions.
- Select an address format by country.
- Validate addresses.
- Define countries for reporting.
- Specify state information.

Pages Used to Set up Country and State Information

Page Name	*Definition Name*	*Navigation*	*Usage*
Country Table - Country Description	COUNTRY_DEFN	EPM Foundation, EPM Setup, Common Definitions, Country and State Info, Country Table, Country Description	Review country descriptions.
Country Table - Address Format	ADDR_FORMAT_TABLE	EPM Foundation, EPM Setup, Common Definitions, Country and State Info, Country Table, Address Format	Select an address format by country. You do so by configuring address fields and field descriptions so that addresses conform to the customary address format of the specified country. After the address format is set, it appears everywhere that the system uses the address subrecord.
Country Table - Valid Address	EO_ADDR_VALIDAT	EPM Foundation, EPM Setup, Common Definitions, Country and State Info, Country Table, Valid Address	Validate addresses by adding valid combinations of address fields.
Country Statistics	COUNTRY_STAT_PNL	EPM Foundation, EPM Setup, Common Definitions, Country and State Info, Country Statistics	Specify countries for reporting.
State	STATE_DEFN	EPM Foundation, EPM Setup, Common Definitions, Country and State Info, State Table, State	Specify state, province, or other geopolitical region information.

Reviewing Country Descriptions

Access the Country Table - Country Description page (EPM Foundation, EPM Setup, Common Definitions, Country and State Info, Country Table, Country Description).

Country Description page

The system displays the country code, description, and a short description for the country.

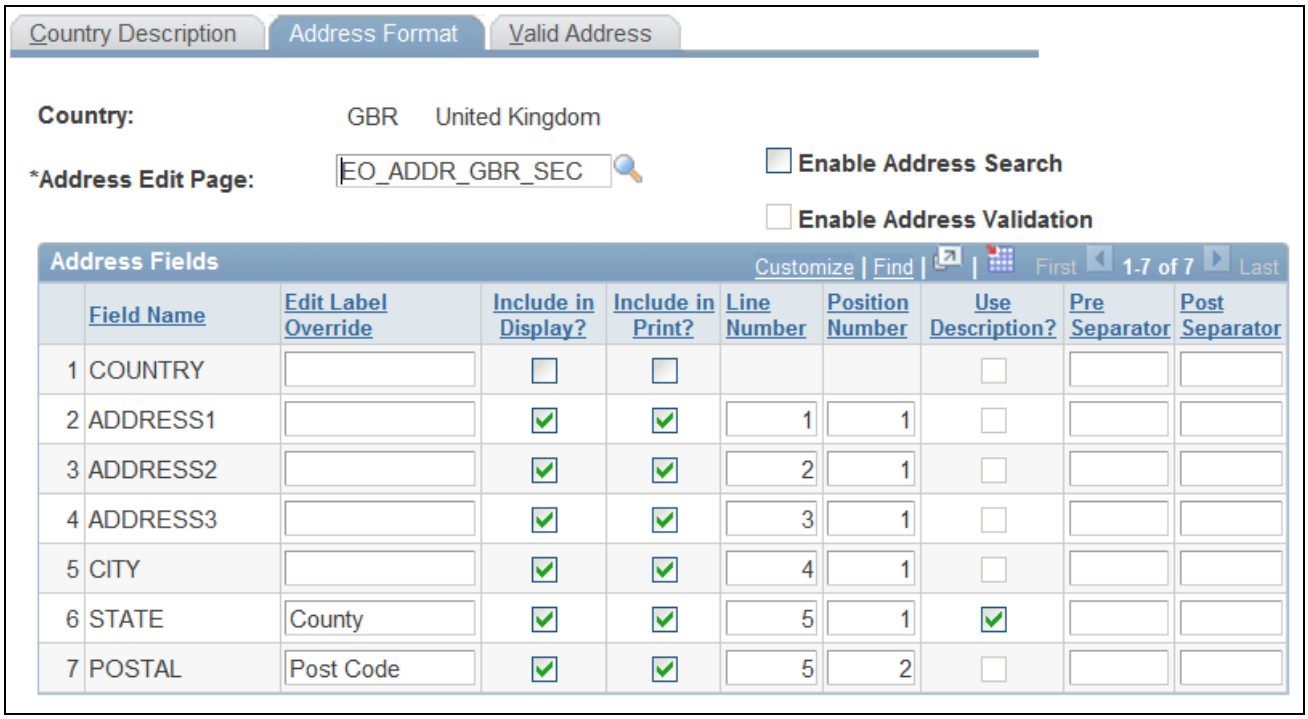
PeopleSoft delivers a fully populated country code table, which is updated as national boundaries and designations change.

2-Char Country Code (two-character country code) Enter a country code. Country codes are informational indicators that record the value-added tax (VAT) registration ID for each country where your company does business.

EU Member State (European Union member state) Select to specify that the country is a member of the European Union.

Selecting an Address Format by Country

Access the Country Table - Address Format page (EPM Foundation, EPM Setup, Common Definitions, Country and State Info, Country Table, Address Format).



Address Format page

Use this page to set up three aspects of the address: editing, displaying, and printing.

The address format that you specify on this page appears everywhere that the address subrecord is used in the system.

Address Edit Page Displays the secondary page used for editing the address. You can create a new secondary page by using PeopleSoft Application Designer. On the new address secondary page that you create, use page fields from the DERIVED_ADDRESS record definition. You then need to add a secondary page control to ADDRESS_SBP that points to your new secondary page. After you complete these steps, the secondary page is accessible in this field.

PeopleSoft delivers default address edit pages that contain the address definitions.

Enable Address Search Select to enable users to search for a valid value. Selecting this check box enables the Used in Search column and the Valid Address page.

Enable Address Validation Available only when Enable Address Search is selected. Select to ensure validation of all values that are selected. When you select this check box, users must select a value from the search list. If this check box is deselected, users can select from the search list or enter a new value.

Address Fields

The system lists available address fields. The settings of these fields are controlled by the selected address edit page.

Field Name	Displays the field options available for the address page.
Edit Label Override	(Optional) Enter an alternative label for the field. The new label is used when prompting for the field. You can customize address formats so that they conform to the address requirements of each location. For example, for a U.S. address, you might change the Postal field label to ZIP Code. Keep in mind the distinctions between county and state: • *County:* The tertiary geopolitical region within a state; the level after country and state. (In the U.K., the level of state is called a county; you would enter such counties in the State field). • *State:* The secondary geopolitical region within a country; a state in the U.S., a province in Canada, a county in the U.K., and a department in France.
Used in Search?	Available only when you select the Enable Address Search check box. Select the fields over which you want users to be able to search.
Include in Display?	Select to activate the corresponding address field as part of the standard address format for the selected country. To remove a field, deselect the check box. When entering addresses, users enter a country code, after which the system updates the page to display the fields appropriate for that country.
Include in Print?	Select to include the field when printing reports.
Line Number and **Position Number**	Enter the physical location of the fields for displaying and printing.
Use Description	Select to display the description for the field value. For example, for addresses in Japan, select this option to display the description of the state rather than the state code, because the code is numeric.
Pre Separator and **Post Separator**	Enter characters to be used surrounding the address field. For example, in the United States, a comma generally follows the city name, such as in *San Francisco, CA.* In India, there are parentheses around the postal code, for example *(123).*

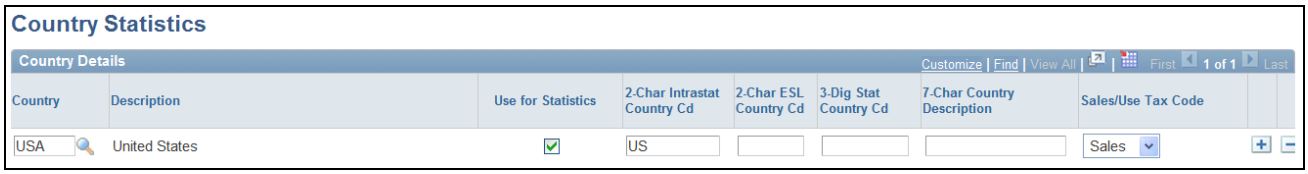
Validating Addresses

Access the Valid Address page (EPM Foundation, EPM Setup, Common Definitions, Country and State Info, Country Table, Valid Address).

To enable this page, select the Enable Address Search check box on the Address Format page, which enables the Used in Search column. The fields that you select to be used in a search appear on the Valid Address page as columns. Enter the valid address combinations that the user can search for and select.

Defining Countries for Reporting

Access the Country Statistics page (EPM Foundation, EPM Setup, Common Definitions, Country and State Info, Country Statistics).



Country Statistics page

PeopleSoft EPM only uses this page to establish the countries for statistical purposes. So most of the information is for informational purposes only, because the reports mentioned are not available through the PeopleSoft EPM database. For more information on the reports mentioned in the following section, refer to the appropriate PeopleSoft Financials application documentation.

Country Select the country on which you would like to report.

Use for Statistics If this check box is selected, the country recognized by the International Standards Organization (ISO) is also recognized by the European statistical offices. If the check box is not selected, the country is recognized only by the ISO. In this case, European statistical offices assume that the country is a part of another country when producing statistical data such as the gross national product (GNP). For example, for the ISO, the principality of Monaco has its own country code, but for statistical purposes, Monaco is assumed to be part of France, and therefore has the same statistical country code as France.

2-Char Intrastat Country Cd (two-character intrastat country code) The country identifier for the European statistical offices. This code often appears on intrastat layout forms.

2-Char ESL Country Cd (two-character ESL country code) Used for countries that are members of the European Union. It usually appears as part of the VAT registration information on the European Sales List (ESL).

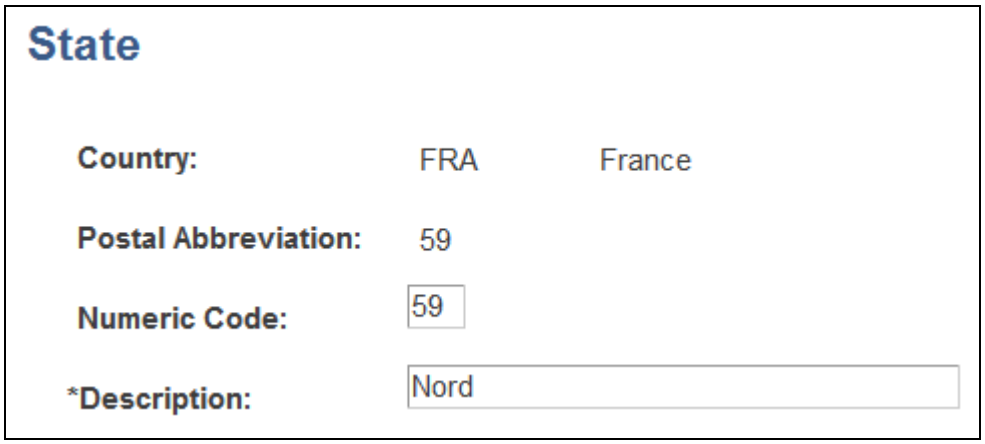
3-Dig Stat Country Cd (three-digit statistical country code) Used as a country identifier for the European statistical offices. It usually appears on the intrastat report to identify source or destination countries to which or from which goods are shipped.

7-Char Country Descr (seven-character country description) Used for the German international Electronic Funds Transfer (EFT) layout. It identifies countries into which or from which electronic funds are sent or received.

Sales/Use Tax Code Informational only. This field can be set to *None*, *Sales*, or *Use*.

Specifying State Information

Access the State page (EPM Foundation, EPM Setup, Common Definitions, Country and State Info, State Table, State).



State page

PeopleSoft delivers a fully populated state code table. PeopleSoft updates the state table as changes occur. This table provides states, provinces, and equivalent geopolitical entities (for example, Dutch communities and French departments) for all supported countries. The codes are based on standard postal codes.

Numeric Code Assign a number to a state or province for statistics and reporting purposes.

Defining Accounting Calendars

This section provides an overview of accounting calendar setup, lists common elements, and discusses how to:

- Define a base calendar.
- Modify or add detail calendars.
- Define summary calendars.
- (Optional) Define business calendars.
- (Optional) Define daily calendars.
- (Optional) Define budget calendars.
- Define calendar frequencies.
- Define frequency details.
- Define timespans.

Understanding Accounting Calendar Setup

In PeopleSoft systems, you can establish an accounting period configuration based on the beginning and ending period dates that you normally use and combine these periods to create accounting calendars. These calendars define the time periods to which you post transactions or create reports for different ledger group and business unit combinations. You select the calendar that defines the periods for a business unit and ledger group combination.

You can maintain an unlimited number of accounting periods over any span of years. You can maintain traditional monthly periods, including an additional adjustment period, or you can define your own periods.

Note. In the PeopleSoft EPM product line, calendars are used mostly for reporting. However, for the system to properly handle data (for example, general ledger data) from your transaction database, you must be sure that the detail calendars match those in your transaction accounting system. You can move calendars from your PeopleSoft transaction database into your PeopleSoft EPM database using PeopleTools such as Data Mover.

The PeopleSoft system supports multiple calendars, so you can keep one calendar for actuals, another for budget and forecast activity, and still others for special reporting needs. Because you store calendars in tables, you can share them across business units, helping to provide consistency in period dates and easing the process of acquisition and consolidation.

PeopleSoft uses the following calendar definition options:

Calendar Builder	Use to create a base calendar from which to create other calendars such as the detail calendar.
Detail Calendar	Define detail calendars that include the number and duration of accounting periods in your fiscal year and the beginning and ending dates for each period. These calendars also identify the adjustment periods for the calendar.
Summary Calendar	Use summary calendars to group detail calendar periods for inquiries and reporting, such as for quarterly reports and semiannual reviews. In this way, your financial information is always ready to be summarized into the timespans that you use most frequently.
(Optional) Business Calendar	Use to create the business or working calendar that identifies holidays and non work days.
(Optional) Daily Calendar	Use for reporting purposes.
(Optional) Budget Period Calendar	Use to manually define fiscal and non fiscal detail budget period calendars.
Calendar Frequencies	Use to define frequency relationships among multiple calendars. You can use frequencies to create multiple relationships among calendars or assign a frequency designation to a calendar. For example, you might have a calendar with a quarterly frequency. This quarterly frequency might be based on the monthly detail calendar.

Timespans Timespans control the number of periods for which data can be extracted from the ledger table. They enable you to easily select and retrieve information for use in allocations and inquiries. Many timespans are expressed relative to the current period, so that they automatically adapt the content of a report to the date it is produced. Others are defined for specific periods. While many commonly used timespans are included in your PeopleSoft system, you can define additional timespans on the TimeSpan page as necessary.

Common Elements Used in This Section

Periods Per FY (periods per fiscal year) Enter the number of periods in an accounting year for a calendar.

Description Enter a description. The description appears on prompt lists, inquiries, and reports.

Descr (description) Enter a detailed description of the calendar.

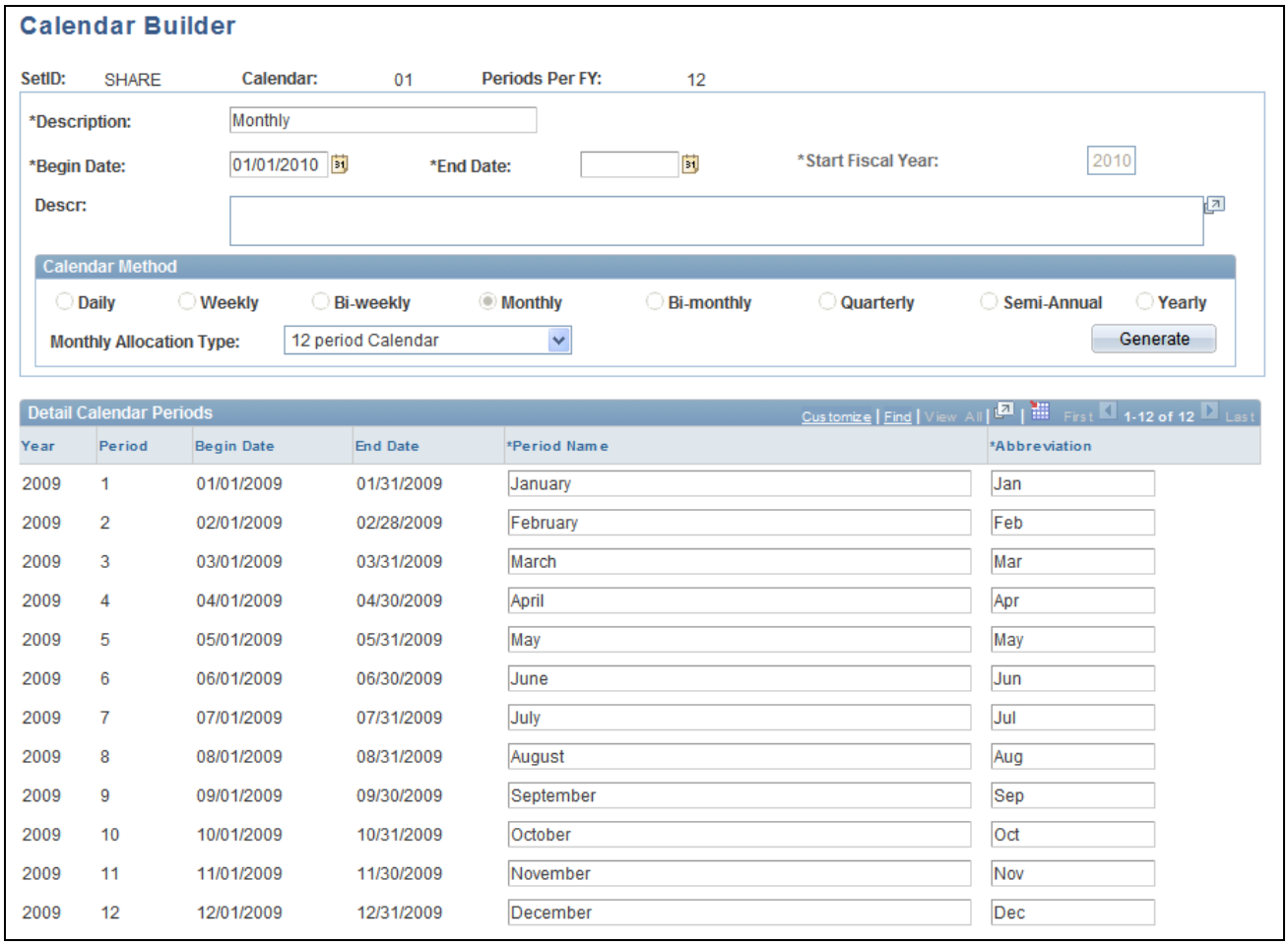
Pages Used to Define Accounting Calendars

Page Name	*Definition Name*	*Navigation*	*Usage*
Calendar Builder	CALENDAR_BUILDER	EPM Foundation, EPM Setup, Common Definitions, Calendars, Calendar Builder	Define a base calendar to be used by other calendars that you create. The calendar that you create is by default a detail calendar.
Detail Calendar	DETAIL_CALENDAR1	EPM Foundation, EPM Setup, Common Definitions, Calendars, Detail Calendar	Modify or add detail accounting calendars that match or differ from your general ledger calendars.
Summary Calendar	SUMMARY_CALENDAR	EPM Foundation, EPM Setup, Common Definitions, Calendars, Summary Calendar, Summary Calendar	Define a summary calendar to group or combine periods from detail calendars.
Summary Budget Period Calendar	SUMMARY_BP_CAL	EPM Foundation, EPM Setup, Common Definitions, Calendars, Summary BP Calendar, Summary Budget Period Calendar	Define a summary budget period calendar to group or combine periods from detail calendars.

Page Name	*Definition Name*	*Navigation*	*Usage*
Business Day Calendar	BUS_CALENDAR	EPM Foundation, EPM Setup, Common Definitions, Calendars, Business Day Calendar	Define a business calendar to specify your normal business week and the holidays observed by your banks.
Daily Calendar	DAILY_CALENDAR	EPM Foundation, EPM Setup, Common Definitions, Calendars, Daily Calendar	Define a daily calendar for reporting purposes.
Budget Calendar	CAL_BP_TBL	EPM Foundation, EPM Setup, Common Definitions, Calendars, Budget Calendar	Define a budget calendar for controlled budget processing.
Frequency Definition	PF_FREQUENCY_DEFN	EPM Foundation, EPM Setup, Common Definitions, Calendars, Frequency, Frequency Definition	Define calendar frequencies. Add and modify frequency definitions.
Frequency Details	PF_FREQUENCY_DTL	EPM Foundation, EPM Setup, Common Definitions, Calendars, Frequency Details	Define frequency details by assigning the frequencies to calendars.
TimeSpan	TIME_SPAN	EPM Foundation, EPM Setup, Common Definitions, Calendars, TimeSpan	Define timespans.

Defining a Base Calendar

Access the Calendar Builder page (EPM Foundation, EPM Setup, Common Definitions, Calendars, Calendar Builder).



Calendar Builder page

Begin Date Enter the beginning date for the calendar. The system uses this date and the end date to create the range of periods for the calendar.

End Date Enter the end date for the calendar. The system uses this date with the begin date to create the range of periods for the calendar.

Start Fiscal Year The system enters a default year for this field based on the year that you enter in the begin date field.

Calendar Method Select one of the following options to determine the number of periods in the calendar: daily, weekly, bi-weekly, monthly, bi-monthly, quarterly, semi-annual, or yearly. For example, a daily calendar has 365 periods and a quarterly will have 4 periods. The system updates to display the number of periods in the Periods Per FY (periods per fiscal year) field.

Monthly Allocation Type Use to select the appropriate period allocations for a monthly calendar. Select from the following: *12 period Calendar*, *13 period Calendar*, *445 Calendar* (4 weeks, 4 weeks, 5 weeks), *454 Calendar* (4 weeks, 5 weeks, 4 weeks), or *544 Calendar* (5 weeks, 4 weeks, 4 weeks).

Generate	Click this option to have the system generate the calendar. The generated calendar periods display begin and end dates in the grid at the bottom of the page.
Period Name and **Abbreviation**	Use these fields to change the generated calendar period name and abbreviation if applicable.

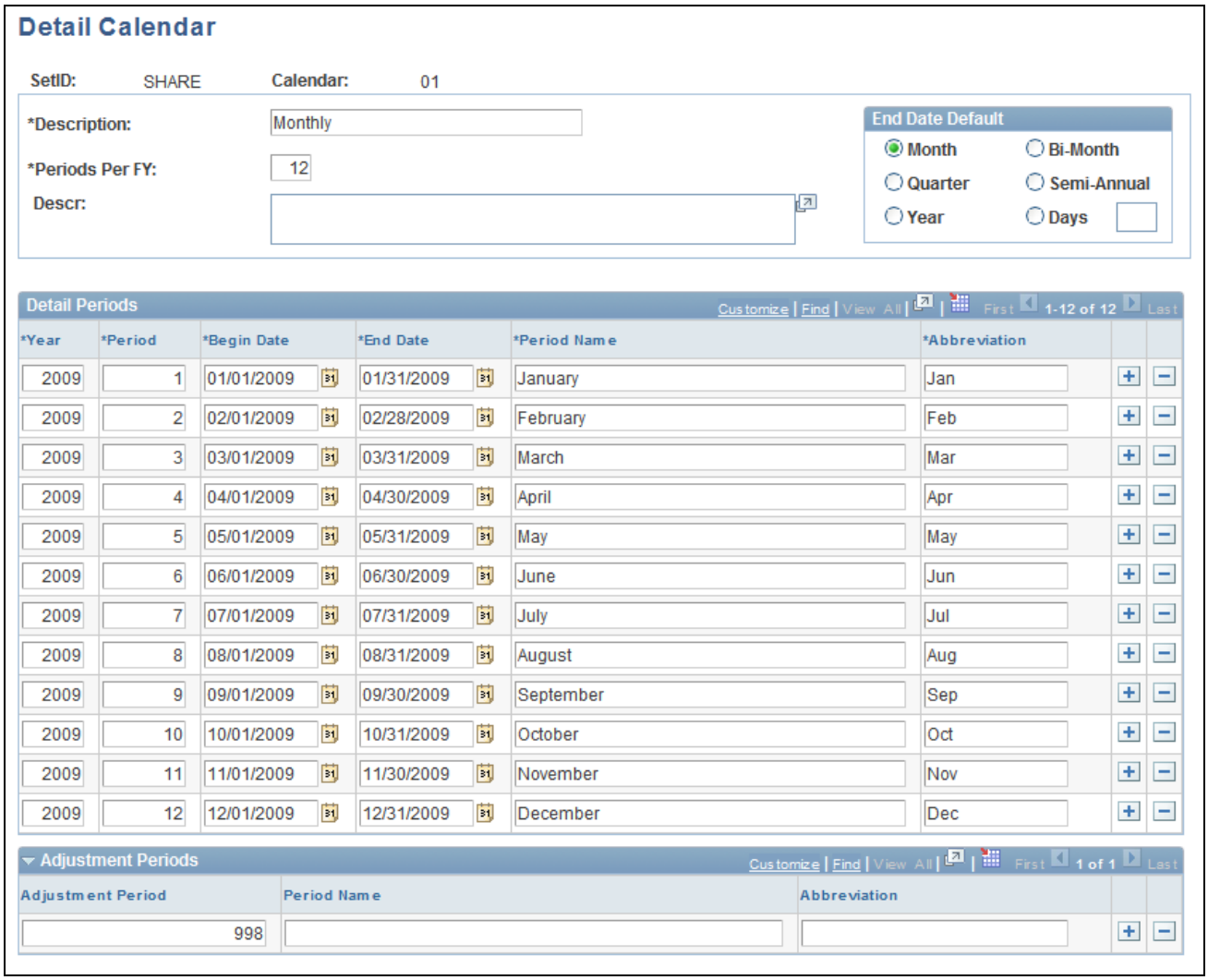
Note. As you define calendars, keep in mind that in all instances in which a PeopleSoft EPM warehouse business unit is related to a general ledger business unit, the calendars for the business units should be the same. Warehouse business units are described elsewhere in this chapter.

After you have saved a calendar using the calendar builder, you can only modify the calendar description or long description on the Calendar Builder page. If you need to modify the saved calendar further, use the other calendar pages, (for example the Detail Calendar page).

By default, a calendar that you create with the Calendar Builder is a detail calendar.

Adding or Modifying Detail Calendars

Access the Detail Calendar page (EPM Foundation, EPM Setup, Common Definitions, Calendars, Detail Calendar).



Detail Calendar page

You create detail calendars using the Calendar Builder. Use this page to modify calendars after they have been created.

Periods per FY (periods per fiscal year) Displays the appropriate number of periods for the calendar. For example, *4* indicates a quarterly calendar.

End Date Default This setting specifies which periodic intervals the system creates. Values are: month, bimonth (bi-monthly), quarter, semi-annual, year, and days.

If you select days, enter the number of days to include in the calendar.

The system uses the end date default setting to populate the subsequent begin and end date values in the grid at the bottom of the page.

Detail Periods Displays the detail periods for the calendar, including the begin and end dates, period name, and abbreviation.

Adjustment Periods Displays any adjustment periods defined for the calendar, including the period name and abbreviation.

When you enter period dates, you can define monthly calendar periods or any fiscal period that matches your accounting calendar (such as weekly or bimonthly) as long as the beginning and ending dates of successive periods don't overlap. Every day of the year must be included in a period; you cannot leave gaps between period dates. Make sure that your detail calendar includes a period for the oldest transaction that you want to enter. After installing your PeopleSoft system, you might want to make this earliest date more restrictive.

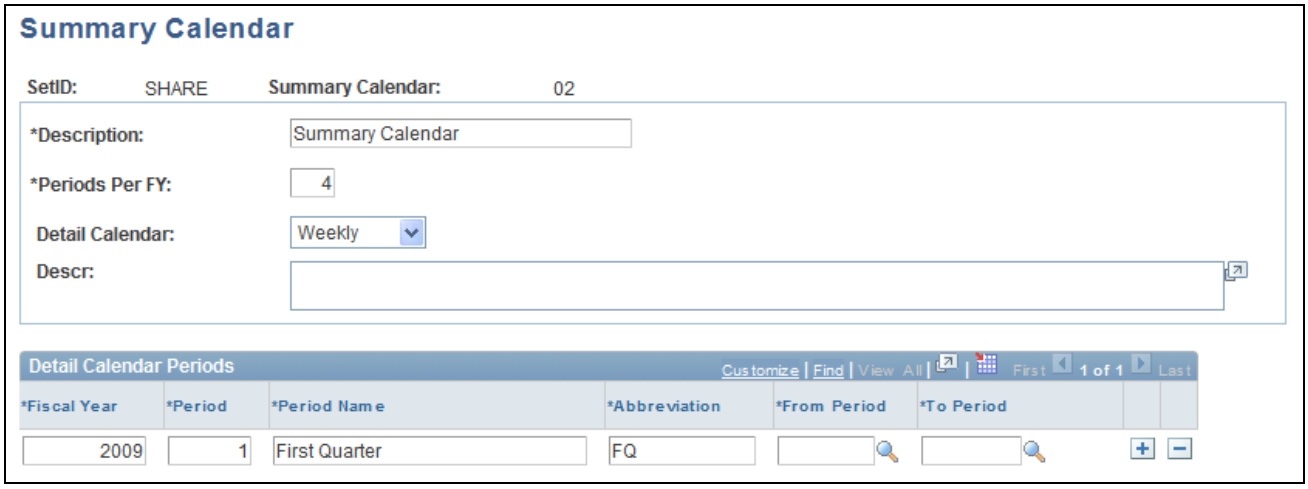
You will need to return to this page to enter ensuing years manually. You can enter several years at a time or treat the task as part of your end-of-year system maintenance.

After you define your detail calendar, you can use it to manage open periods for the generation of journals. You can also use it to define the periods that store summarized results in a summary ledger.

Note. The MODEL calendar delivered with your system contains data from 1957 to 2025. You can use the MODEL calendar rather than entering your own data; if you do so, make a copy of MODEL first, and make changes to the copy.

Defining Summary Calendars

Access the Summary Calendar page (EPM Foundation, EPM Setup, Common Definitions, Calendars, Summary Calendar, Summary Calendar).



Summary Calendar page

Consider the following when setting up summary calendars. A quarterly summary calendar, for example, might have four periods, each consisting of three periods from a monthly detail calendar. A summary calendar can also combine cumulative detail calendar accounting periods to create year-to-date balances. Summary calendars are especially useful for determining how your models will roll up for reporting purposes. As with detail calendars, you can include as many fiscal years as you need in one summary calendar.

Enter a description and periods per fiscal year to specify the number of periods in the accounting year for this calendar.

Detail Calendar Select the ID of the detail calendar on which this calendar is based. Every summary calendar must be based on a detail calendar.

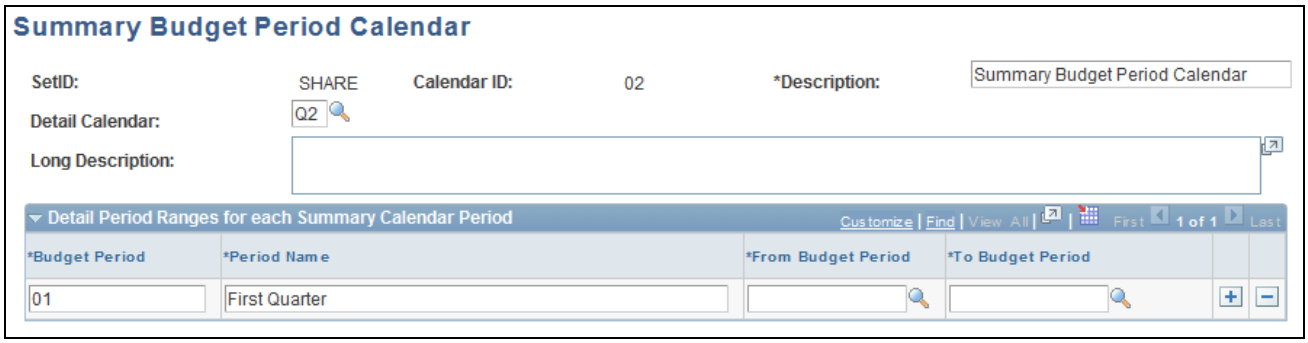
Supply the fiscal year, period, period name, and abbreviation.

From Budget Period and **To Budget Period** — Enter the detail calendar periods to be summarized in each summary period.

Note. You can define as many years on a calendar as necessary. Be sure to include any years that you use to store historical information.

Defining Summary Budget Period Calendars

Access the Summary Budget Period Calendar page (EPM Foundation, EPM Setup, Common Definitions, Calendars, Summary BP Calendar, Summary Budget Period Calendar).

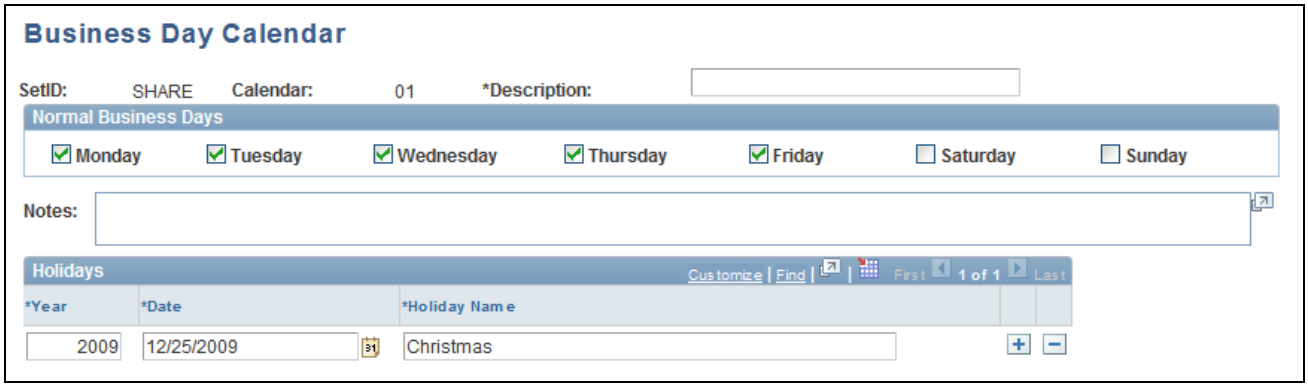


Summary Budget Period Calendar page

Summary calendars are especially useful for determining how your models will roll up for reporting purposes. Enter a description, budget period, and from/to periods to specify the number of budget periods in the accounting year for this calendar.

Defining Business Calendars (Optional)

Access the Business Day Calendar page (EPM Foundation, EPM Setup, Common Definitions, Calendars, Business Day Calendar).



Business Day Calendar page

Normal Business Days — Select the business days of the week to include.

Notes Enter any notes to further describe this calendar.

In the grid at the bottom of the page enter a row for each holiday on which you know businesses won't operate.

Year and **Date** Specify the year and date for any holidays that you include.

Holiday Name Enter the name of the holiday on which your business does not operate.

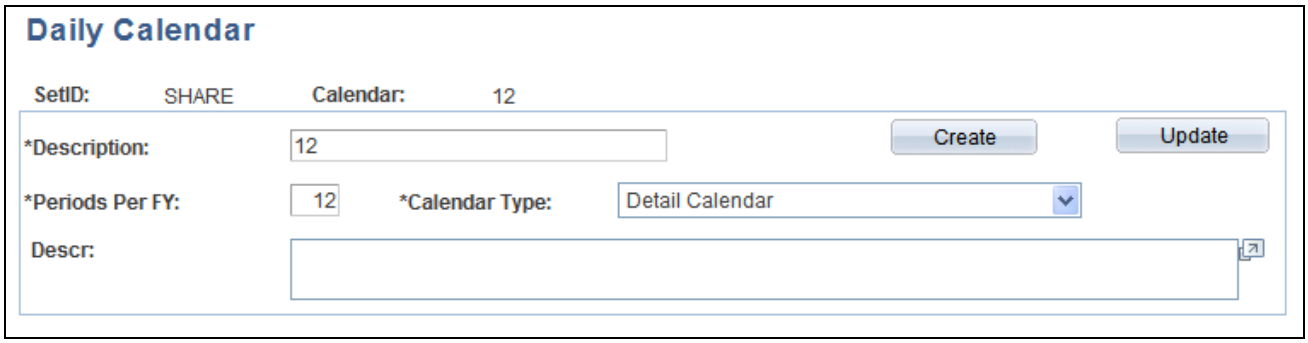
In multinational corporations, you accommodate the various locations and different holidays observed by defining as many business calendars as you need for each setID.

Note. Business calendars determine the number of workdays in each month, for reporting purposes only.

Note. If you use PeopleSoft Asset Liability Management, Funds Transfer Pricing, or Risk-Weighted Capital, you must create a business day calendar to define your holidays.

Defining Daily Calendars (Optional)

Access the Daily Calendar page (EPM Foundation, EPM Setup, Common Definitions, Calendars, Daily Calendar).



Daily Calendar page

Create Click to access the Create Detail Periods secondary page, on which you can enter the fiscal year and start date for creating daily periods.

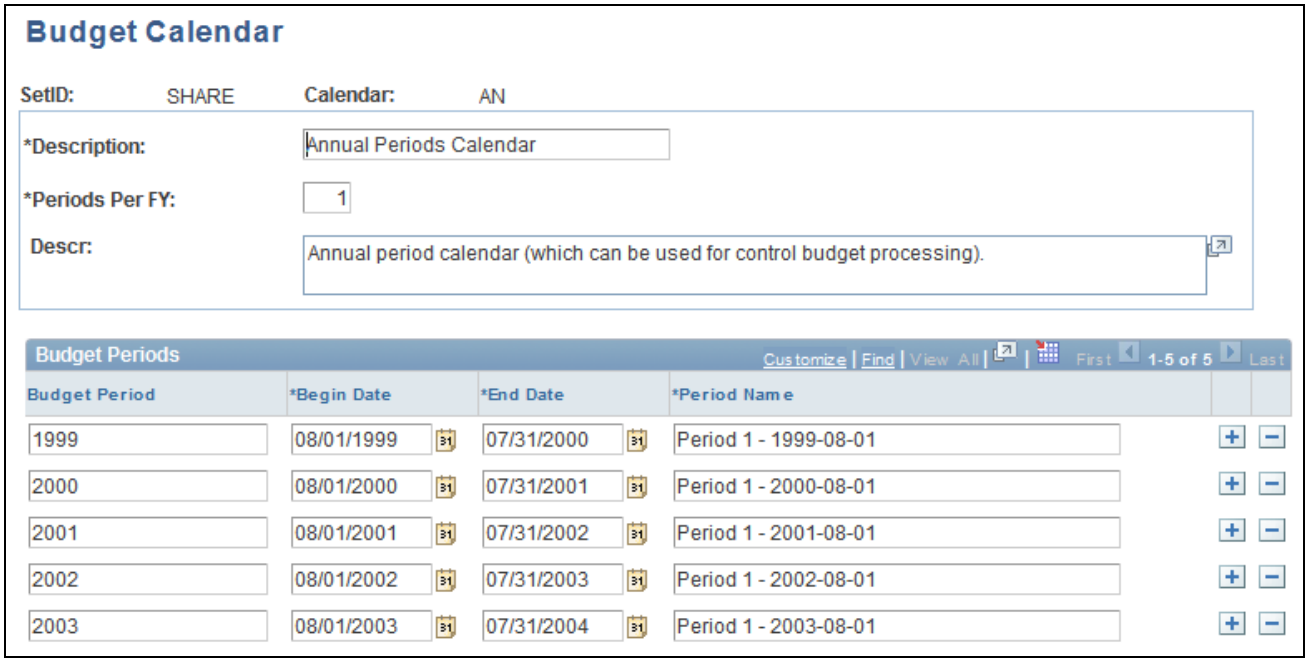
Update Click to access the Create Detail Periods secondary page, on which you can enter a new fiscal year.

Remember to set up a calendar for each fiscal year. To display the detail periods for existing calendars, click the Update button.

Note. For leap years, you must change the periods per fiscal year value from 365 to 366 to create the last period for the leap year (for example, December 31, 2004).

Defining Budget Calendars (Optional)

Access the Budget Calendar page (EPM Foundation, EPM Setup, Common Definitions, Calendars, Budget Calendar).



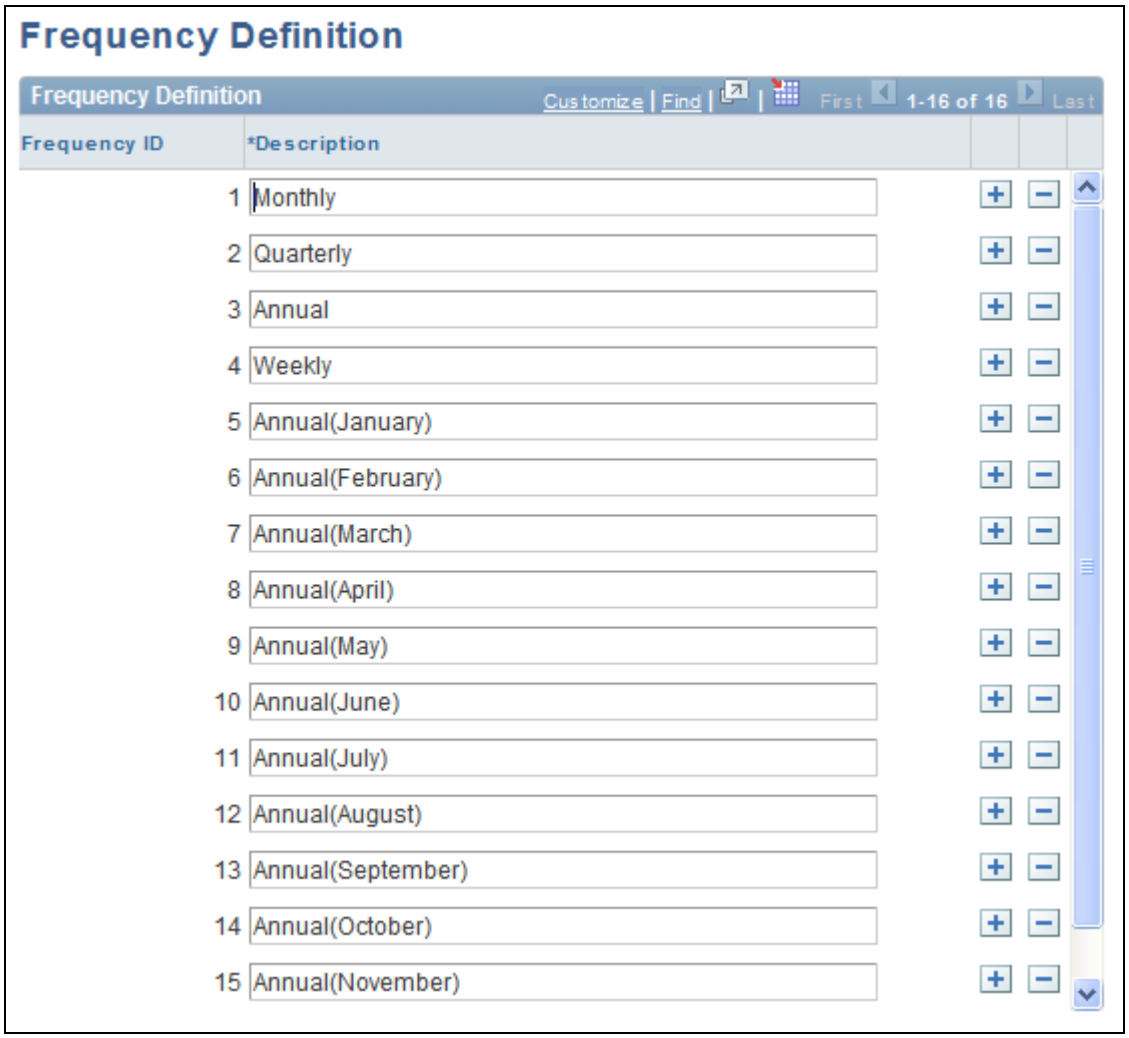
Budget Calendar page

For budget calendars, you might want to enter a descriptive calendar ID such as *AN* for an annual calendar budget.

Enter the budget periods, begin and end dates, and period names.

Defining Calendar Frequencies

Access the Frequency Definition page (EPM Foundation, EPM Setup, Common Definitions, Calendars, Frequency, Frequency Definition).



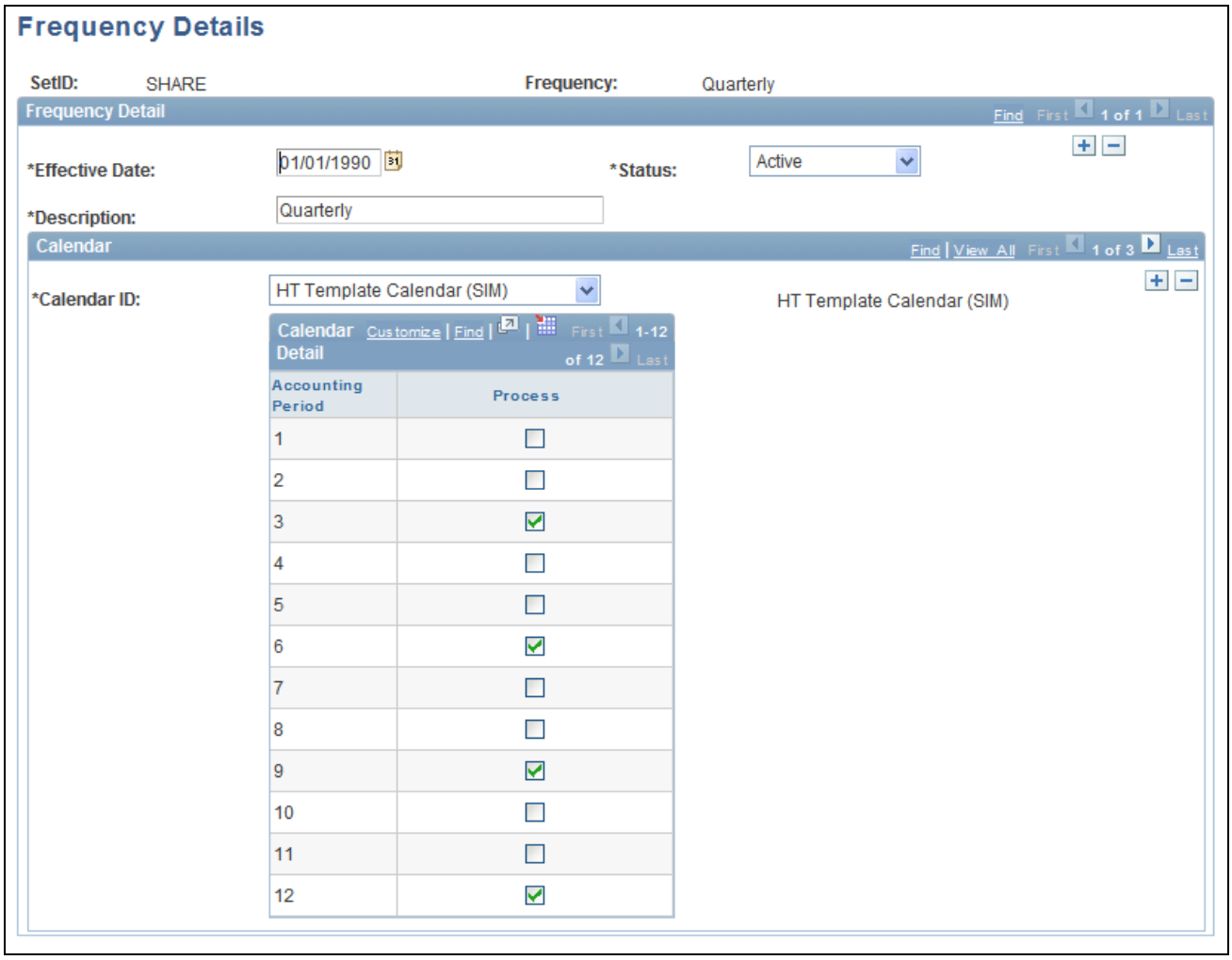
Frequency Definition page

Define frequencies by adding a row with a description or by updating an existing description.

Note. The frequencies you define on this page will comprise the list of valid values in the Frequency Details page.

Defining Frequency Details

Access the Frequency Details page (EPM Foundation, EPM Setup, Common Definitions, Calendars, Frequency Details).



Frequency Details page

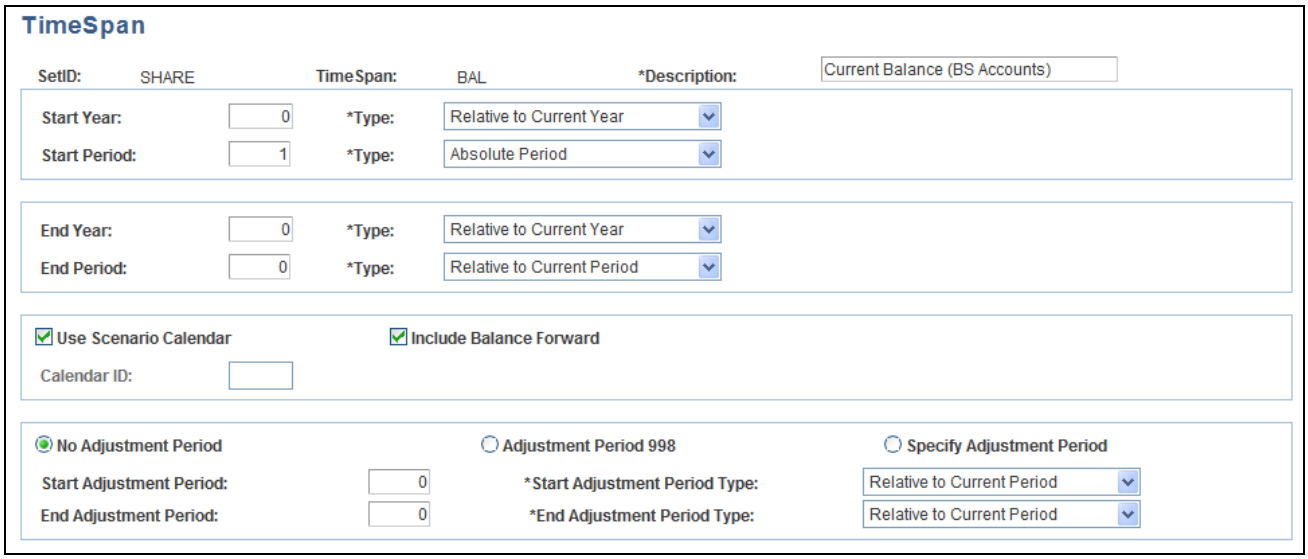
Calendar ID Select a calendar. The calendar ID that you select represents the calendar that you want to associate with the frequency. You can have more than one frequency defined for a single calendar. The system updates to display calendar detail.

You can also specify all frequencies that are associated with any other frequency definition for this frequency group. You can define frequency relationships across multiple calendars.

Process Select for any of the accounting periods that you want to process.

Defining Timespans

Access the TimeSpan page (EPM Foundation, EPM Setup, Common Definitions, Calendars, TimeSpan).



Time Span page

Start Year and **Start Period**
Enter the appropriate value. The value that you enter depends on the type that you select.

End Year and **End Period**
Enter the appropriate value. The value that you enter depends on the type that you select.

Type
Select from the following:

Absolute Period or *Absolute Year*: You can enter any year or period that has been defined in the timespans calendar.

Relative to Current Period or *Relative to Current Year*: Enter *0* as the period or year to select the current year or period, *-1* to indicate the previous year or period, *1* to indicate the next, and so forth.

Use Scenario Calendar
Select this check box to use the specified scenario calendar with this timespan. This enables you to have a scenario-based timespan.

Include Balance Forward
Select this check box if you use nVision for the consolidation ledger or any ledger with balance forward accounts. If you select this check box, nVision will compute ledger balance.

Calendar ID
Select the calendar that you want to use with this timespan.

Note. Balance forward accounts store balance forward amounts. For example, in your regular accounting system you might want to specify asset, liability, and equity accounts as balance forward accounts, but not revenue or expense accounts. The account balance forward function rolls forward the accounts on any date specified for flexible year-end processing.

See Also

PeopleSoft Enterprise Global Consolidations 9.1 PeopleBook

Setting Up the Gregorian Calendar [EPM Warehouses]

The Calendar dimension represents calendars that fall into two broad functional categories. The Gregorian calendar (the calendar that is commonly used) has one row for each day. The business calendar represents the functionality the PeopleSoft provides to define various types of business calendars, such as detail calendars, summary calendars, budget calendars, and so on.

Gregorian calendar setup is a onetime setup as part of the MDW setup. You can generate the Gregorian calendar for a range of years. For each day in the year range, the Gregorian calendar has one row. (You do not define business calendars in the MDW; their definitions are imported from source systems and the OWE).

To set up the Gregorian calendar, use the Day Dimension (D_DAY) component and the Time Dimension (D_TIME) component.

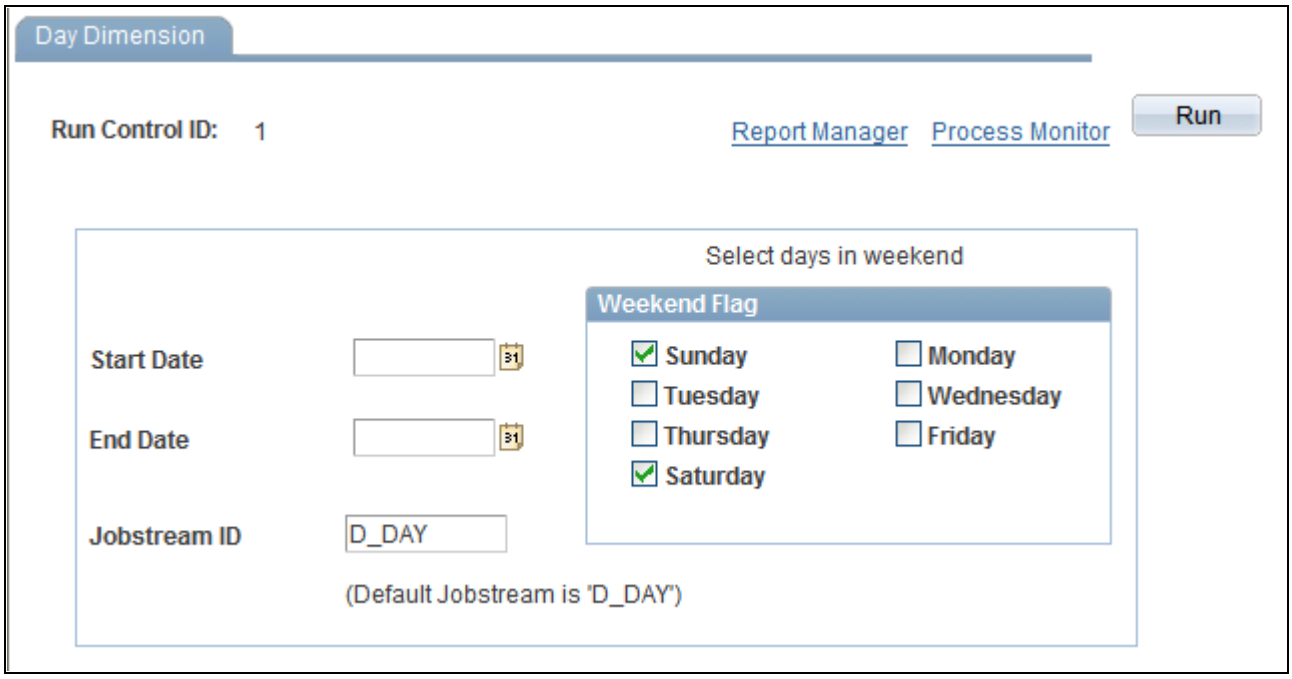
This section discusses how to set up the Gregorian Calendar for the EPM warehouses.

Pages Used to Set Up the Gregorian Calendar

Page Name	*Definition Name*	*Navigation*	*Usage*
Day Dimension	D_DAY	EPM Foundation, EPM Setup, Common Definitions, Calendars, Gregorian Calendar, Gregorian Calendar Dimension, Day Dimension	Populate the Day dimension.
Time Dimension	D_TIME	EPM Foundation, EPM Setup, Common Definitions, Calendars, Gregorian Calendar, Time Dimension, Time Dimension	Populate the Time dimension.

Populating the Day Dimension

Access the Day Dimension page (EPM Foundation, EPM Setup, Common Definitions, Calendars, Gregorian Calendar, Gregorian Calendar Dimension, Day Dimension).



Day Dimension page

Start Date Enter the beginning date for this calendar.

End Date Enter the ending date for this calendar.

Jobstream ID Enter *D_DAY*.

Weekend Flag Check the days that represent weekend days.

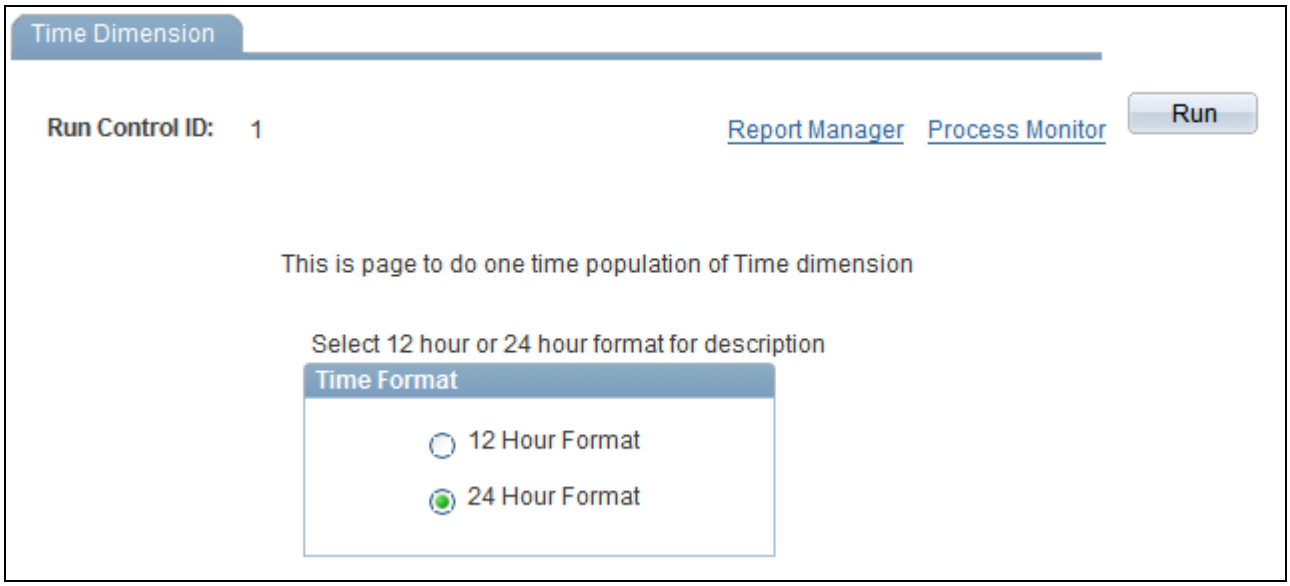
Retry Enabled This field will be available for future use.

Save the page and click Run.

Click the Process Monitor link to check the status of application engine program. After a period of time, it should display the status as *Success.*

Populating the Time Dimension

Access the Time Dimension page (EPM Foundation, EPM Setup, Common Definitions, Calendars, Gregorian Calendar, Time Dimension, Time Dimension).



Time Dimension page

Time Format Select either 12 Hour Format or 24 Hour Format.

Save the page and click Run.

Click the Process Monitor link to check the status of application engine program. After a period of time, it should display the status as *Success.*

Setting Up Time Zones [EPM Warehouses]

EPM extract, transform, and load (ETL) jobs use the date time stamp in conjunction with source records to perform incremental loads. However, data can be extracted from multiple source transaction systems, each of which might reside in different database servers that use different time zones and different date and time stamps. This can lead to mismatched dates and times between the source and the EPM database. However, EPM target warehouse tables may include a source time zone field for records that have a date-time or time value depending upon the design and requirement.

When a target warehouse table contains a time zone field, prepackaged ETL jobs populate it with your source time zone values. Not all target warehouse tables contain a time zone field and only one source time zone value can be populated in the target warehouse table.

Populating warehouse target tables with source time zone values is the first step in ensuring the synchronicity of source and EPM time zones. However, you are responsible for converting the source time zone value to the EPM database time zone.

Note. You only need provide time zone data if you are using *Enterprise Learning Management* reports.

Defining Units of Measure

Units of measure (UOM) determine how resources are quantified. Each resource must be associated with a standard unit of measure. Standardization helps you to control the units that appear in reports and enables you to use the PeopleSoft automatic conversion features. ETL jobs move unit of measure data from your source transaction system to EPM target warehouse tables. You need only use the Unit of Measure PIA page to redefine or modify your existing unit of measure data.

This section discusses how to:

- Import UOM values from your source.
- Display and modify UOMs.

Importing Unit of Measure Values from Your Source

EPM enables you to extract and consolidate source data from various source transaction systems. As such, certain source systems might contain UOM values that are different from those offered in EPM. This causes mismatched UOM values between the source and the EPM database. However, EPM target warehouse tables include source UOM definitions and conversion rate fields for all records that store UOM data. Prepackaged ETL jobs populate these fields with your source UOM definitions and conversion rate values.

PeopleSoft-delivered ETL jobs populate the following target warehouse tables with UOM values:

- UNITS_TBL
- UNITS_CVT_TBL
- INV_ITEM (for Supply Chain Management only)
- UOM_TYPE_INV (for Supply Chain Management only)

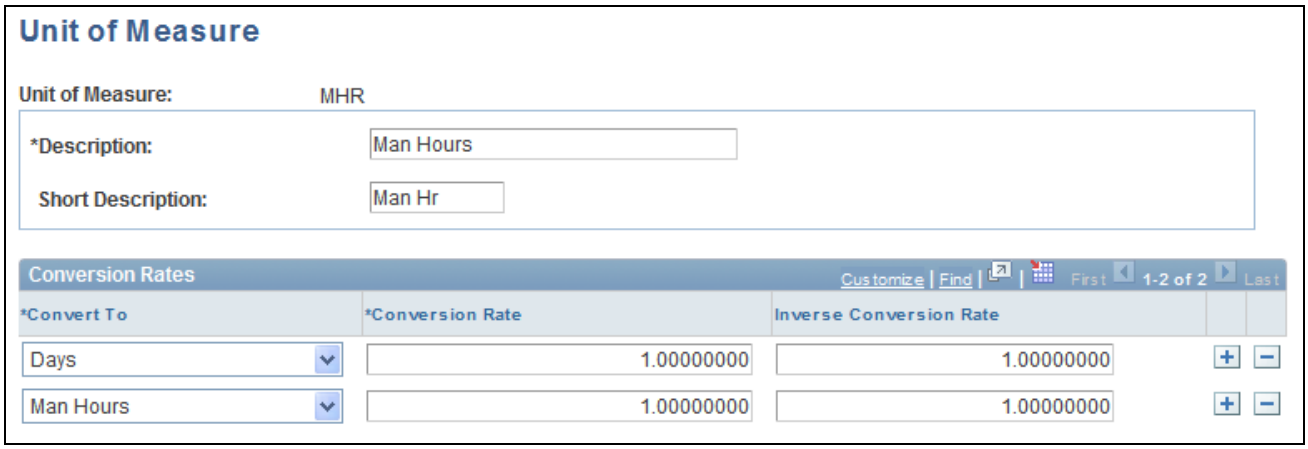
Populating warehouse target tables with source UOM values is the first step in ensuring the synchronicity of source and EPM UOM values. However, you are responsible for converting the source UOM value to the EPM database UOM.

Page Used to Modify UOMs

Page Name	*Definition Name*	*Navigation*	*Usage*
Unit of Measure	UNITS_TBL1	EPM Foundation, EPM Setup, Common Definitions, Unit of Measure	Display and modify units of measure to determine how resources are quantified.

Displaying and Modifying UOMs

Access the Unit of Measure page (EPM Foundation, EPM Setup, Common Definitions, Unit of Measure).



Unit of Measure page

Convert To, Conversion Rate, and **Inverse Conversion Rate**

Specify the conversion rules of a particular UOM.

For example, by specifying a conversion from cubic feet to cubic yards and an appropriate conversion rate, when you enter a resource amount in cubic feet, the system automatically converts the amount to cubic yards. Automatic conversion is a key feature of PeopleSoft statistical accounting. It enables you to enter resource amounts in whatever unit is readily available or convenient at the time, and converts the units to the required standard. The system also automatically calculates the inverse conversion rate.

Defining and Maintaining Dimensions

This section provides an overview of dimensions and discusses how to:

- Define dimensions.
- Define dimension details for an Operational Warehouse - Enriched (OWE) warehouse table.
- Define dimension details for a Multidimensional Warehouse (MDW) table.

Understanding Dimensions

Dimension metadata is associated with specific record metadata and its fields. For example, you can define the dimension DEPARTMENT that references the record DEPARTMENT_TBL and the DEPT_ID column in that record. Dimension metadata captures additional column, key, and security information that is not included with a standard datamap, such as alternate key fields and dimension security. A single dimension can be defined for both an OWE and MDW tables, enabling you to use the same dimension name for both table types.

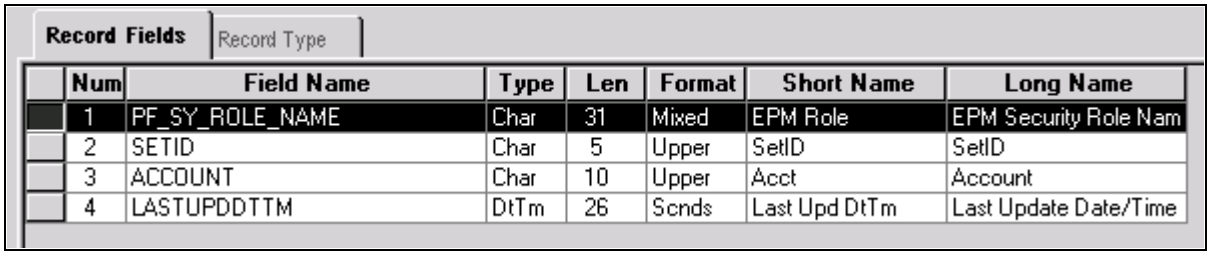
Dimension metadata is used by the EPM Warehouses, applications, security, and KPI manager.

Dimension Security

Because EPM is delivered with no security restrictions, dimensions are also delivered unsecured. Before you can grant a user access to a dimension you must first indicate to the system that a particular dimension requires securing. Dimensions that are not secured are classified as public, or unsecured. All EPM users can view these objects.

You specify dimensions that require securing using the Dimension page. After you specify a dimension to secure you must associate that dimension with a *security join table* to complete dimension security. Security join tables are EPM database tables that store the security profiles for users along with the corresponding dimension values for which they have access. During security processing, a security join table acts as a lookup. For example, when a user is trying to access a row of data, the SQL that processes this request uses the security join table to identify the user and her access to the particular row of data.

A security join table must be created for every dimension that you plan to secure. Each security join table should match the key structure of the dimension table for which it is defined. Each row in a security join table identifies a user or security role and his access to a specific dimension value. A user who is granted access to multiple values in a single dimension table has several rows in the security join table. In the event that a user has access to an entire dimension, you can insert a single row designated *all* and prevent the table from ballooning in size with several rows of data. A user that is granted access to multiple dimensions appears in several security join tables. The following is an example of a security join table.



Record Fields | Record Type

Num	Field Name	Type	Len	Format	Short Name	Long Name
1	PF_SY_ROLE_NAME	Char	31	Mixed	EPM Role	EPM Security Role Nam
2	SETID	Char	5	Upper	SetID	SetID
3	ACCOUNT	Char	10	Upper	Acct	Account
4	LASTUPDDTTM	DtTm	26	Scnds	Last Upd DtTm	Last Update Date/Time

Account security join table

The security join table model is better than a single security output table for two main reasons: Processing smaller tables is more efficient when you are inserting or deleting data, or querying the table to determine access privileges, and modeling individual security tables enable you to be in sync with the anticipated migration to data objects in future releases.

See Chapter 6, "Setting Up EPM Security," Defining Dimension and Metric Security, page 128.

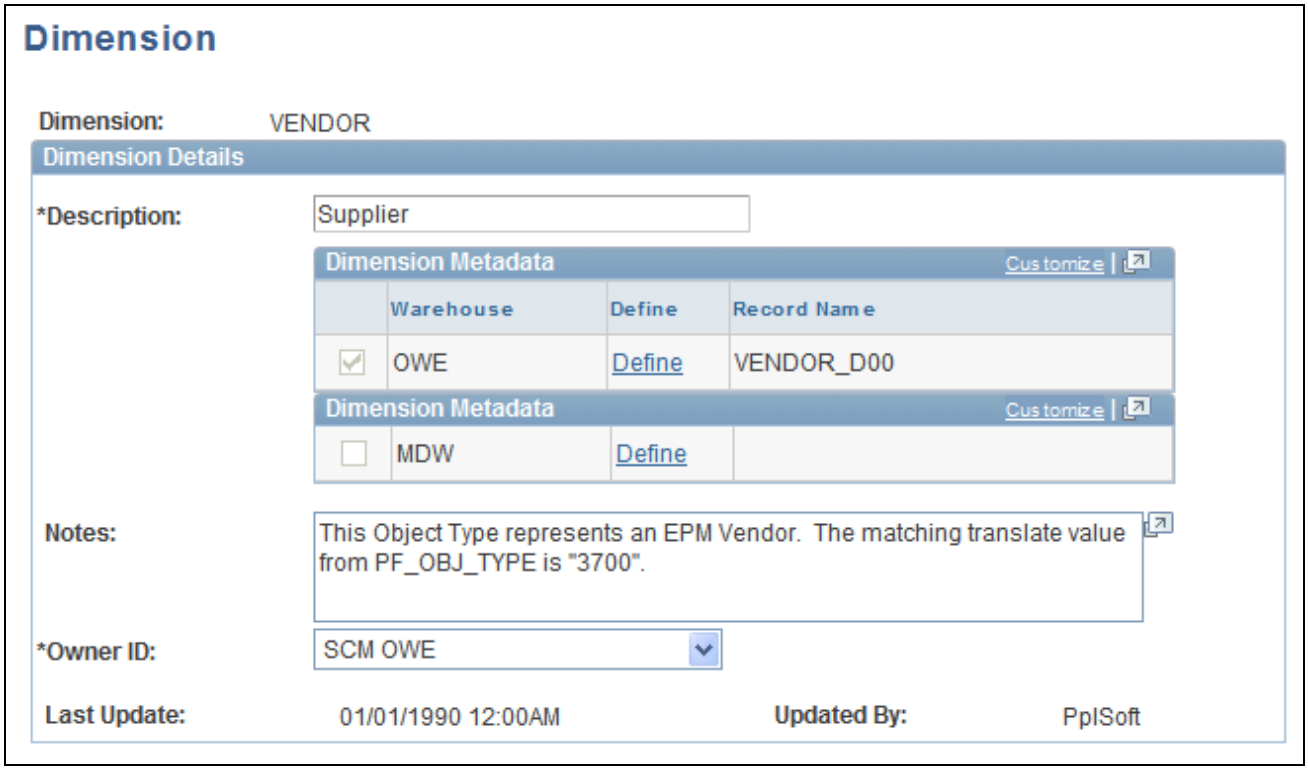
Pages Used to Set Up Dimensions

Page Name	*Definition Name*	*Navigation*	*Usage*
Dimension	PF_DIM_DEFN	EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Define Dimension	Define dimension metadata.
OWE Detail for Dimension	PF_DIM_OWE_DETL	Click the Define link on the Dimension page	Define dimension metadata details for an OWE dimension and apply security parameters.

Page Name	Definition Name	Navigation	Usage
MDW Detail for Dimension	PF_DIM_MDW_DETL	Click the Define link on the Dimension page	Define dimension metadata details for an MDW dimension and apply security parameters.

Defining Dimensions

Access the Dimension page (EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Define Dimension).



Dimension page

Warehouse Indicates the warehouse layer that is associated with each dimension type.

Define Click to access the OWE Detail for Dimension or MDW Detail for Dimension page, depending on which dimension type you are defining.

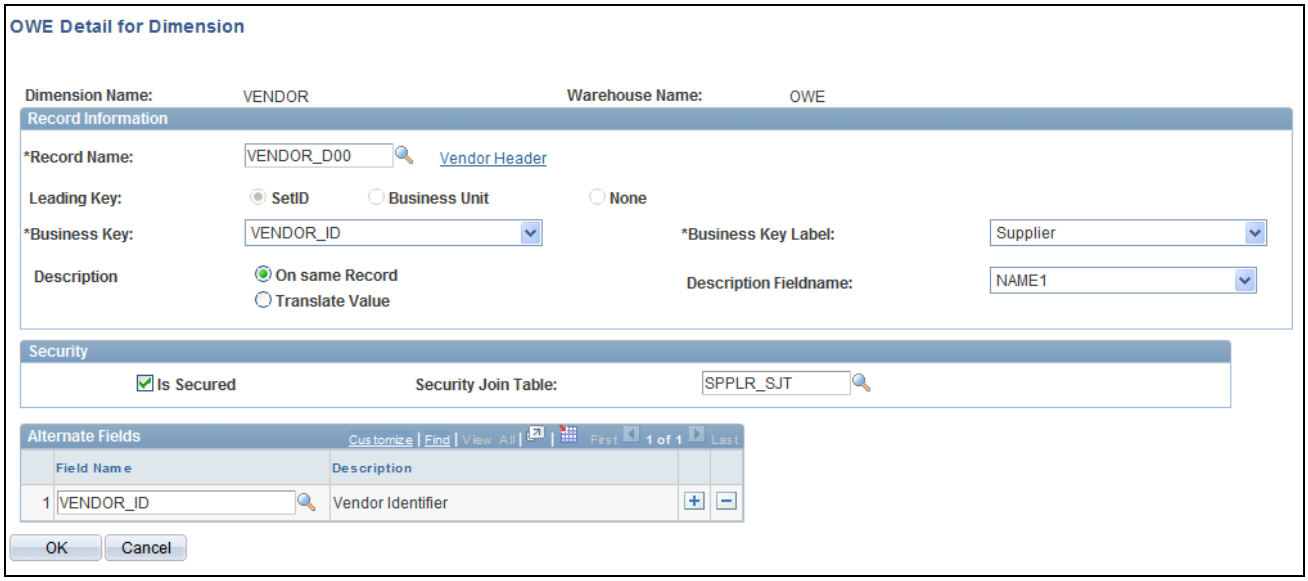
Record Name Displays the name of the record that is associated with a particular dimension.

Owner ID Select an owner ID for this dimension.

The owner represents an EPM functional area (such as ABM or Budgeting) that is associated with the dimension. Assign an owner ID to help organize and group the metadata—making locating and auditing easier.

Defining Dimension Details for an OWE Warehouse Table

Access the OWE Detail for Dimension page (Click the Define link on the Dimension page).



OWE Detail for Dimension page

Record Information

Record Name Select the record that you want to associate with this dimension.

Leading Key Select a leading key for set processing.

SetID indicates that the leading key is a setID.

Business Unit indicates that the leading key is a business unit.

None indicates that there is no leading key.

Business Key Select a business key to serve as the primary key for the selected dimension.

Business Key Label Select a label for your business key.

Description Select a description for the business key field.

On Same Record indicates that the description is available in the record. If you select this option, specify a value for the Description Fieldname field.

Translate Value indicates that the description is available through the translate table.

Description Fieldname Select the field which contains the description of the business key field.

Create Record Metadata Select to access the Record Metadata page and define new record metadata to associate with this dimension.

If you select an existing record from the prompt, this field displays the name of the record metadata.

Security

Is Secured Select this check box to indicate whether this dimension is secured.

If you secure this dimension, you must specify its related security join table in the Security Join Table field.

Security Join Table Select a security join table to associate with this dimension.

The security join table is an EPM table that stores the security profiles for users along with the corresponding dimension values to which they have access. Only Dimensions that are associated with security join tables are presented to a security administrator when security is defined.

Alternate Fields

Field Name Select an alternate name for your dimension fields.

Alternate field names share the same IDs as the primary field and can be used by different fact tables to join to the same ID.

Specifying alternate field names is optional.

Defining Dimension Details for an MDW Table

Access the MDW Detail for Dimension page (Click the Define link on the Dimension page).

The MDW Detail for Dimension page contains the same fields as the OWE Detail for Dimension page with the addition of the following three fields.

Surrogate Key Select a surrogate key for this dimension.

MDW dimensions are keyed by surrogate key, and not business unit.

Source ID Select the source transaction system that is associated with the data in the record.

Prompt View Enter a key to map the surrogate key to the business key.

This field enables you to select data using meaningful keys instead of surrogate keys, which are generated numbers.

Defining Operator Defaults

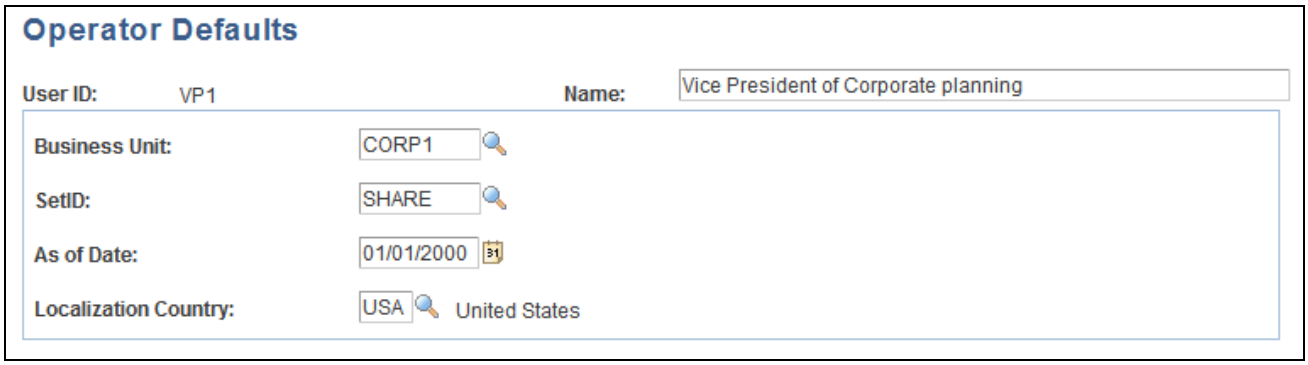
Operator defaults enable you to assign a default business unit and setID for each user per country. You specify operator defaults on the Operator Defaults page.

Page Used to Specify Operator Defaults

Page Name	*Definition Name*	*Navigation*	*Usage*
Operator Defaults	OPR_DEF_TABLE_FS1	EPM Foundation, EPM Setup, Installation Analysis and Options, Personal Defaults, Operator Defaults	Specify default user preferences for PeopleSoft users.

Specifying Default User Preferences

Access the Operator Defaults page (EPM Foundation, EPM Setup, Installation Analysis and Options, Personal Defaults, Operator Defaults).



Operator Defaults page

Business Unit and **SetID** The values that you select become the user's default values in the business unit and setID search fields.

Localization Country Enter the default country for this user ID.

Note. You can control the business unit, setID, and address format default values by using the Operator Defaults page.

Archiving EPM Data

This section provides an overview of EPM data archive and discusses how to define a warehouse archive project.

Understanding EPM Data Archiving

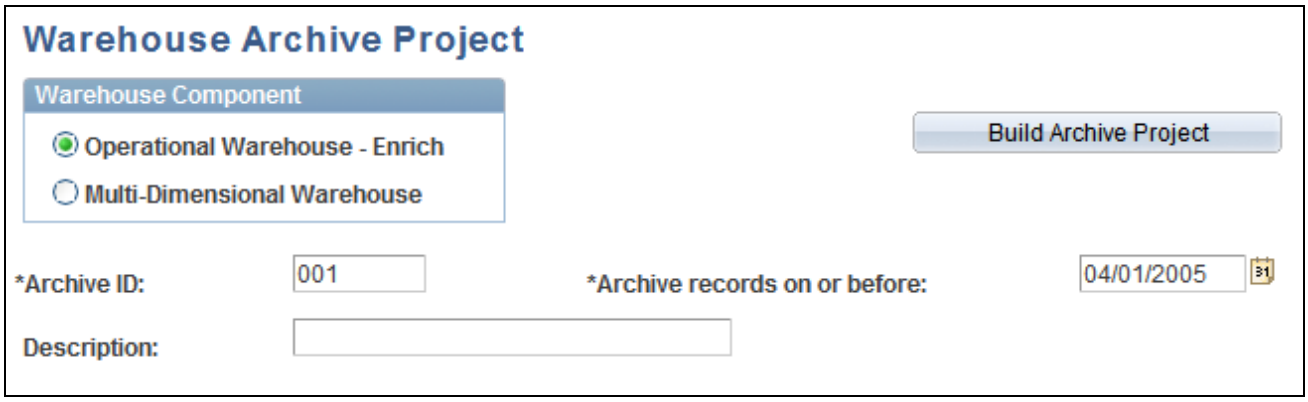
You can use the warehouse archive project functionality to archive your OWE or MDW data. The Warehouse Archive Project links to the PeopleTools archive tool when you have defined your selection criteria. The archived data is deleted from the database and saved to a flat file.

Page Used to Archive EPM Data

Page Name	*Definition Name*	*Navigation*	*Usage*
Warehouse Archive Project	PF_ARCH	EPM Foundation, Foundation Metadata, Other Metadata Operations, Build Archive Project, Warehouse Archive Project	Archive your EPM data.

Defining a Warehouse Archive Project

Access the Warehouse Archive Project page (EPM Foundation, Foundation Metadata, Other Metadata Operations, Build Archive Project, Warehouse Archive Project).



Warehouse Archive Project page

Warehouse Component Select the component to archive: Operational Warehouse - Enrich (OWE) or Multi-Dimensional Warehouse (MDW).

Archive ID Enter an archive ID for your project.

Archive records on or before	Select the date. Only records with a date equal to or before the date that is specified will be archived.
Build Archive Project	Click this button to build the archive project. The system accesses the PeopleTools, Archive Data component. You will need to manage the rest of the archival process from the PeopleTools pages. For information about this component, see your PeopleSoft PeopleTools documentation.

Note. Make sure the Archive to Flat File check box is selected on the PeopleTools Record Criteria page (ARCH_PROJ).

Setting Up Chart Viewing and Printing Results [Analytical Applications]

Some analytical applications use charts. To view these charts, the AVS charting tool must be installed. In addition, complete the following procedure to set up required options:

1. Navigate to EPM Foundation, EPM Setup, Installation Analysis & Options, Installation Options, Web Services.
2. Select the Chart Server check box.

   This enables you to use the basic charting features.
3. To enable applets, select the Applets check box.

   Note that applets download code to the client workstation.

You can also print any of the pages that display results. To produce an optimal printout, set your internet browser to print background colors and images.

In Internet Explorer, follow these steps to set this option:

1. Select Tools, Internet Options.
2. Select the Advanced tab.
3. Scroll through the list of settings to view the Printing options.
4. Select the Print background colors and images check box, and click OK.

Setting Up Ledger-Based nVision Reporting for EPM

This section discusses how to set up ledger-based nVision reporting over EPM ledger tables.

Understanding Ledger-Based Reporting and Setup in EPM

nVision is a reporting tool that can retrieve information from a PeopleSoft database and place that data into a Microsoft Excel spreadsheet for further analysis. nVision selects data from a PeopleSoft database using a *query* or a *ledger*. nVision query-based reporting uses PS query to extract information and can be used with the EPM database. However, nVision ledger-based reporting must be modified to extract data from the EPM database. This is due to the manner in which ledger-based reporting extracts data and the unique structure of EPM ledger tables.

nVision ledger-based reporting uses the LEDGER key to distinguish sets of ledger data and identify which rows should be extracted from ledger tables. However, the LEDGER key does not exist in the EPM ledger tables (such as PF_LEDGER_F00). Instead, EPM ledger tables use the key, PF_SCENARIO_ID, to distinguish sets of data. Consequently, the ledger-based reporting process must be modified to accommodate the PF_SCENARIO_ID used by EPM ledger tables.

To incorporate the PF_SCENARIO_ID key into the nVision ledger-based reporting process, you must create a view of your EPM ledger tables and map the PF_SCENARIO_ID column to the LEDGER column in the view. EPM delivers a view of the PF_LEDGER_F00 table called PF_LEDGER_VW, so you do not need to create one. You only need to create additional ledger views for those ledger tables associated with EPM analytical applications (for example, GC_LEDGER_F00 for Global Consolidations). For more information about application ledger tables that require a view, see the PeopleBook for your PeopleSoft EPM analytical application.

In addition to mapping keys with a ledger view, you must also set up related nVision ledger pages in PIA. The EPM nVision PIA pages populate tables that are used during the ledger-based reporting process (for example, PF_LED_DEFN). When nVision processes a request, it looks for the SET_ID and LEDGER keys located in the PF_LED_DEFN table. The PF_LED_DEFN table is tied to the PF_LED_TEMPLATE table via the LEDGER key, and that connection is used to determine the related data associated with a particular ledger view and PF_SCENARIO_ID. Most of this information is defined in the EPM nVision PIA pages, which are described in the subsequent sections.

See *PeopleSoft Enterprise PeopleTools PeopleBook: PS/nVision*

See *PeopleSoft Enterprise PeopleTools PeopleBook:PeopleSoft Application Designer Developers Guide*

Ledger-Based Reporting Setup

The following steps are required to implement ledger-based nVision reporting for EPM:

1. Create a ledger template for each ledger view in your database.

   The Ledger Template page populates the LED_TMPLT_TBL table, which stores the ledger view information, used to tie the LEDGER key to the PF_SCENARIO_ID key.

2. Create a detail ledger definition for each scenario.

   The Detail Ledger page populates the LED_DEFN_TBL table, which stores the LEDGER and SET_ID keys, used for lookup by nVision reporting.

3. Create a ledger group for each detail ledger.

   The Group Definition page enables you to create a ledger group for one or more detail ledgers and specify base currency for those ledgers.

4. Create a ledger group for each detail ledger.

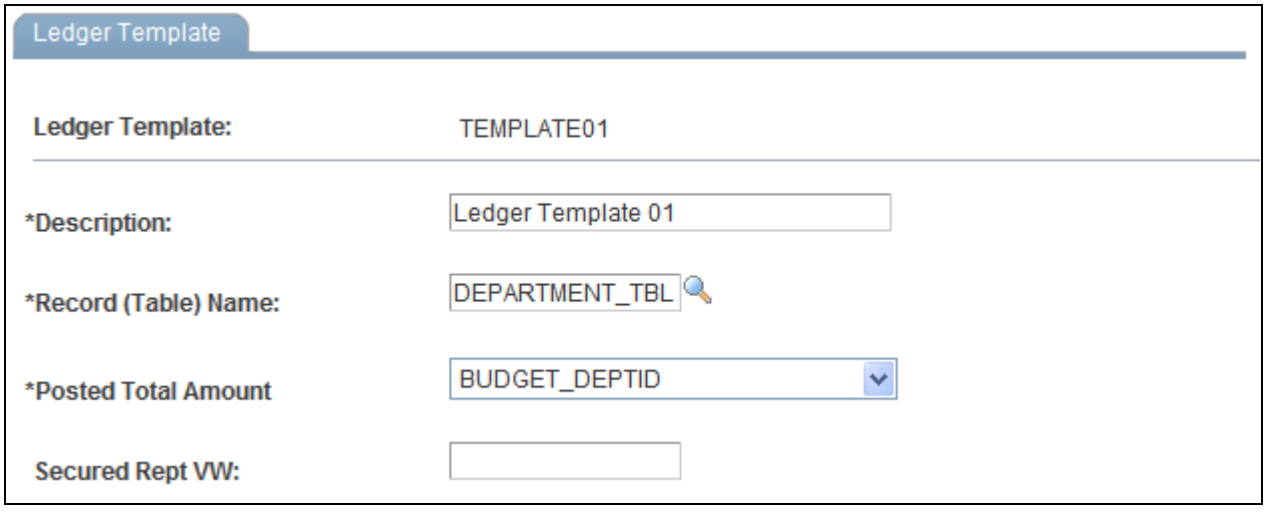
   The Ledgers for a Unit Definition page enables you to associate a calendar ID with a ledger group.

Pages Used to Set Up nVision Ledger-Based Reporting in EPM

Page Name	*Definition Name*	*Navigation*	*Usage*
Ledger Template	NVS_LED_TMPLT	EPM Foundation, EPM Setup, Ledger Setup, nVision Ledgers, Ledger Template, Ledger Template	Set up ledger templates for your views.
Detail Ledger	NVS_LEDGER_DETAIL	EPM Foundation, EPM Setup, Ledger Setup, nVision Ledgers, Detail Ledger, Detail Ledger	Create detail ledgers for your scenarios.
Group Definition	NVS_LEDGER_GROUP	EPM Foundation, EPM Setup, Ledger Setup, nVision Ledgers, Ledger Group, Definition	Create a ledger group for one or more detail ledgers and specify base currency for those ledgers.
Ledgers for a Unit Definition	NVS_BU_LED1	EPM Foundation, EPM Setup, Ledger Setup, nVision Ledgers, Ledgers for a Unit, Definition	Associate a calendar ID with a ledger group

Defining a Ledger Template

Access the Ledger Template page (EPM Foundation, EPM Setup, Ledger Setup, nVision Ledgers, Ledger Template, Ledger Template).



Ledger Template page

Record (Table) Name Select a view to associate with this template.

Note. The record name you select must match the view defined for the EPM ledger record.

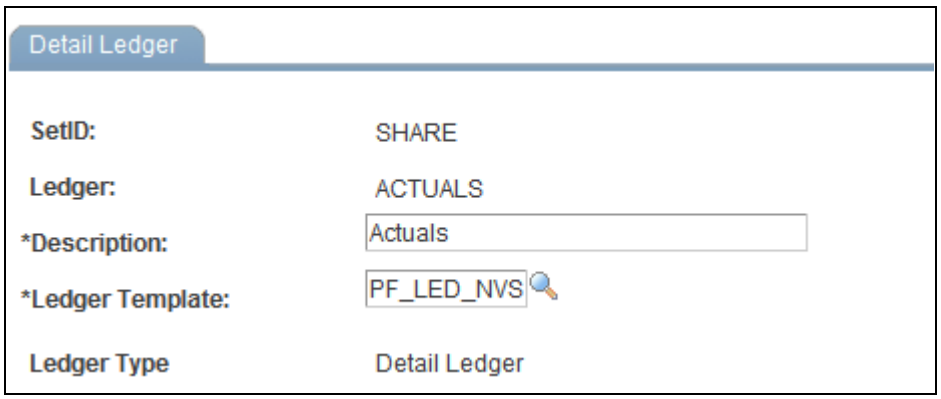
Posted Total Amount Select the name of the record column that holds your posted total amount value.

Secured Rept VW (Secured Report View) Enter the name of an alternate secured reporting view for the template.

This field enables you to use an alternate view for reporting and overrides the view specified in the Record (Table) Name field. You can create a ledger template using a specific record, but allow the reporting of that record be accomplished through a different view.

Defining a Detail Ledger

Access the Detail Ledger page (EPM Foundation, EPM Setup, Ledger Setup, nVision Ledgers, Detail Ledger, Detail Ledger).



Detail Ledger page

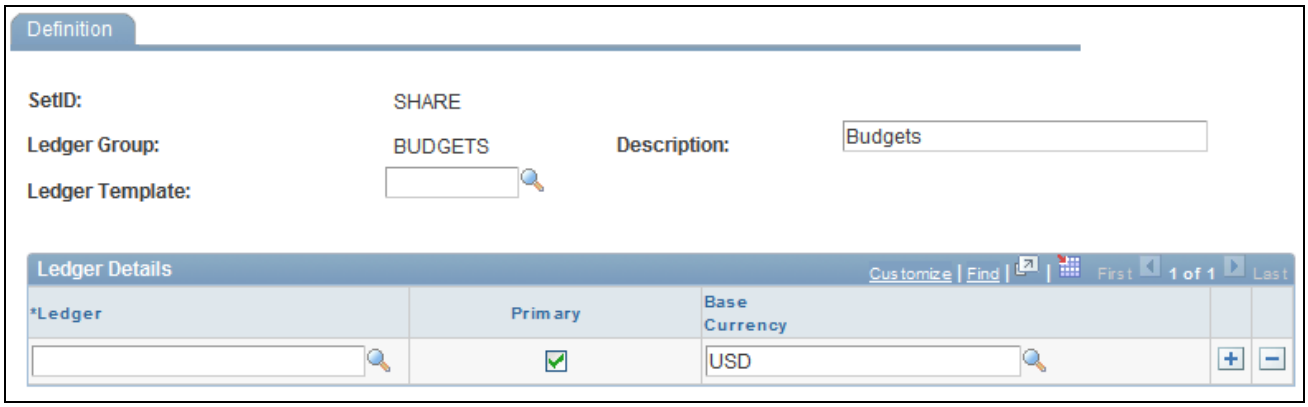
Ledger Displays the ledger record associated with the specified setID.

Ledger Template Select the ledger template to associate with the specified ledger record.

Ledger Type Displays the ledger record type

Defining a Ledger Group for Detail Ledgers

Access the Ledger Group Definition page (EPM Foundation, EPM Setup, Ledger Setup, nVision Ledgers, Ledger Group, Definition).



Ledger Group Definition page

Ledger Template Select a ledger template to associate with the ledger group.

Ledger Details

Ledger Select a ledger scenario to associate with the ledger group.

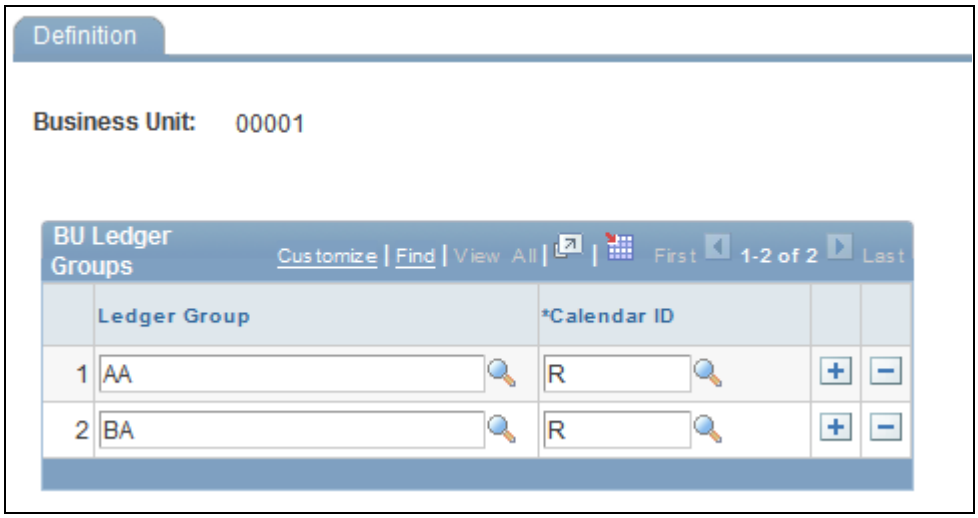
Primary Select this check box to indicate the ledger scenario is the primary ledger for this ledger group.

Note. You must specify at least one primary ledger record.

Base Currency Select the base currency for the selected ledger record.

Defining a Calendar for a Ledger Group

Access the Ledgers for a Unit Definition page (EPM Foundation, EPM Setup, Ledger Setup, nVision Ledgers, Ledgers for a Unit, Definition).



Ledgers for a Unit Definition page

The Ledgers for a Unit Definition page enables you to associate a calendar ID with a ledger group.

Note. The calendar ID should match the calendar ID that is specified on the Warehouse Business Unit Scenario Definition page.

Chapter 5

Setting Up Currency Rules for EPM

PeopleSoft enables you to manage financial information in multiple currencies. PeopleSoft provides specific input, processing, and reporting features that support the European Common Currency (euro), currency conversions, remeasurement, revaluation, translation, and a complete audit trail of all multi-currency processing.

You can define and maintain tables that describe currency codes, exchange rates, market rates, and currency rate types. All PeopleSoft products use the same market rate and currency pages and tables, which enables you to administer centralized currency controls throughout the integrated product lines.

This chapter provides an overview of multiple currency processing concepts and discusses how to:

- Set up EPM currency tables
- Set up market rates
- Define currency quotations
- Establish market rates
- Calculate currency rates
- Configure currency precision

Note. The multi-currency processing setup tasks documented in this chapter are common to both the EPM Warehouses and the Analytical Applications. However, additional multi-currency processing setup tasks are required for the EPM Warehouses and the Analytical Applications:

EPM Warehouses: see the chapter entitled 'Implementing Currency Conversion' in your specific EPM warehouse PeopleBook (for example, the *PeopleSoft Enterprise Campus Solutions Warehouse PeopleBook*).

Analytical Applications: see the chapter entitled 'Setting Up and Running Currency Conversion for the OWE' in this PeopleBook.

Understanding EPM Multiple Currency Processing Concepts

Before you begin to process multiple currencies, you should understand the concepts behind multi-currency processing in EPM.

Currency and Calculation Types

PeopleSoft software uses terminology associated with currency that is consistent with generally accepted accounting principles and the Financial Accounting Standards Board (FASB) accounting standards.

*Currency*refers to the denomination of a monetary transaction. PeopleSoft applications use a currency code (CURRENCY_CD) to identify and track individual currencies. Although the system does not require it, you should use International Standards Organization (ISO) currency codes. PeopleSoft applications have no limit on the number of currencies that you can use.

Important currency terms are:

Base currency	Primary currency in use for a business unit. Each business unit must have one base currency. This is usually the local currency for the organization, but accounting rules or other circumstances might dictate that it be different. In PeopleSoft EPM, you assign a base currency to each warehouse business unit.
European Common Currency (euro)	Common currency adopted by participating European countries (effective January 1, 1999). PeopleSoft applications enable you to use *triangulation* to convert currencies that newly participate in but are not yet replaced by the euro.
Foreign currency	Any currency that a business unit uses other than its base currency for doing business is referred to as a foreign currency. Some foreign currencies are used for reporting only, some are for input only, but most are available for both input and reporting.
Functional currency	Defined in FASB 52 as "currency of the primary economic environment in which a foreign entity operates." It is not an interchangeable term for base currency. When the functional currency differs from the base currency, FASB 52 requires an additional translation (called *remeasurement*) from base to functional currency.
Reporting currency	Used for financial reports such as consolidated financial statements.

Important currency calculation types are:

Exchange rate	Expresses the value of one currency in terms of another. Recognized types of exchange rates include *spot* (immediate), *current, negotiated* (discount and premium forward rates), *average,* and *historical* rates. PeopleSoft applications support any number of exchange rate types.
Conversion	Exchange of one currency for another currency. In PeopleSoft applications, *conversion* refers to the expression of the value of foreign currency transactions in the base currency.
Market rate	Encompasses a number of different rate types including exchange rates, interest rates, stock exchange indexes, economic indicators, and commodity prices.

Remeasurement	Changing the unit of measure from the base currency of a business unit to its functional currency. This is required whenever a business unit's books are maintained in a currency other than the functional currency.
Translation	Expressing ledger balances in terms of another currency, such as when balances maintained in the base or functional currency are restated in a different reporting currency. In the case of translation, gains and losses are recognized solely from fluctuations in the exchange rate.

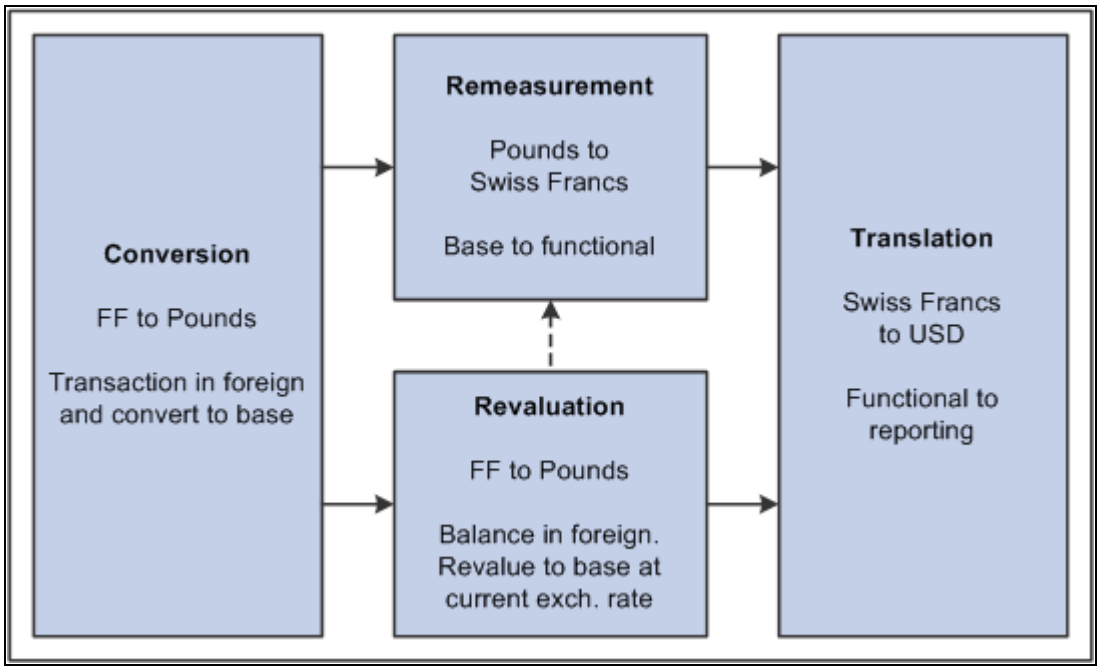
Currency Precision

Currency dictates the *precision* of monetary amounts. For example, United States dollar amounts have two digits to the right of the decimal and Japanese yen have none. The system addresses currency precision as follows:

- PeopleSoft software provides currency-sensitive amount fields with a standard length of 23.3, or 23 digits to the left of the decimal point and 3 digits to the right of the decimal point.
- By default, the system rounds all currency-sensitive amount fields to the currency precision of the associated currency. This action is a PeopleSoft PeopleTools option that you can deactivate.
- All numbers on SQR reports are currency-sensitive. For reporting with Crystal and PS/nVision, the display is equal to the field precision, but you can increase the number of decimal places.

Revaluation

When you adjust the base currency value of balance sheet accounts that are maintained in a foreign currency, this is called *revaluation.* You generally perform revaluations at the end of each accounting period to reflect the actual base currency value of assets and liabilities as exchange rates fluctuate between the base and foreign currencies. You make adjusting entries to the accounts that are being revalued with an offsetting entry to a revaluation gain or loss account. The gain or loss account is sometimes referred to as an unrealized exchanged gain or loss.



Revaluation

In this example, a London-based subsidiary of a Swiss company records a purchase made in Mexican pesos. The Swiss company is owned by a United States corporation. The following table correlates the terminology and the currencies:

Foreign currency	MXN (Mexican Peso)
Base currency	GBP (British pounds)
Functional currency	CHF (Swiss francs)
Reporting currency	USD (United States dollars)

Currency as a ChartField

You get the best results when you record multi-currency transactions with a currency ChartField. This approach enables you to record multiple currencies in the same ledger and reinforces the concept of a ledger's role as a record for an entire category of information (such as actuals, budgets, forecasts, or commitments).

Differentiating Between the Currency Conversion Process of the EPM Warehouses and the Analytical Applications

Separate currency conversion processes are required in EPM, one for the *EPM Warehouses* and one for the *Analytical Applications.* After performing the basic multi-currency processing setup tasks in this chapter, you must perform additional multi-currency processing setup tasks that are specific either to the EPM Warehouses or the Analytical Applications. It is important to understand the difference between the two currency conversion processes, as they are quite different.

The following table describes the differences between the EPM Warehouse and Analytical Application currency conversion processes:

Subject	***Analytical Application Currency Conversion***	***EPM Warehouse Currency Conversion***
Technology Platform	Based on Application Engine (AE) technology for seamless integration with application processing that is also based on AE technology.	Based on ETL technology.
Set Processing	Business unit is required in set processing.	Business unit is optional in set processing. This allows for a single conversion process to convert all transaction amounts for global reporting.
Business/Conversion Rules	Currency conversion involves complex rules for compliance reporting and simulation. The complex rules are stored in various EPM metadata.	Currency conversion is used to convert monetary amounts to a common currency for trend analysis. Trend analysis requires a simple currency conversion rule based on an exchange date and rate type that does not require extensive rule setup.

The remaining currency conversion setup tasks required for the EPM Warehouses can be found in your specific EPM warehouse PeopleBook (for example, *PeopleSoft Enterprise Campus Solutions Warehouse PeopleBook*).

The remaining currency conversion setup tasks required for the Analytical Applications can be found in this PeopleBook.

See Chapter 18, "Setting Up Business Rules for the Operational Warehouse - Enriched," Setting Up and Running Currency Conversion, page 443.

Setting Up EPM Currency Tables

Currency code pages define each currency that you use. To meet your multicurrency requirements, PeopleSoft supports the euro and delivers the Currency Code table with many common ISO standard currencies. The table also supports the ISO standard of zero, two, and three decimal positions.

PeopleSoft-delivered ETL jobs move currency code data from your source transaction system to EPM target warehouse tables. You need only use the currency code PIA pages to redefine or modify your existing currency code data.

This section discusses how to:

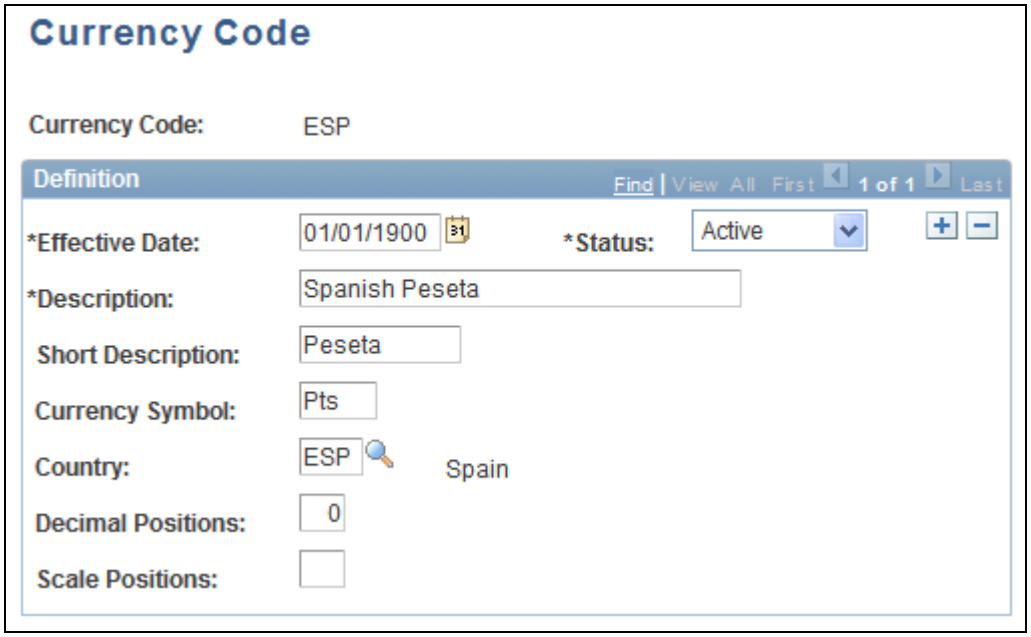
- Maintain currency codes.
- Update the status of a euro currency code.

Pages Used to Maintain Currency Tables

Page Name	*Definition Name*	*Navigation*	*Usage*
Currency Code	CURRENCY_CD_TABLE	EPM Foundation, EPM Setup, Common Definitions, Currencies and Rates, Currency Codes	Maintain currency codes and the currency precision by currency.
Currency Codes (Euro)	CURRENCY_EURO	EPM Foundation, EPM Setup, Common Definitions, Currencies and Rates, Currency Code (Euro)	Update the status of a euro currency code.

Maintaining Currency Codes

Access the Currency Code page (EPM Foundation, EPM Setup, Common Definitions, Currencies and Rates, Currency Codes).



Currency Code page

Currency Symbol PeopleSoft delivers many currencies with a currency symbol such as *$* for Australian dollar (AUD) or *?* for British pound (GBP). You can enter new symbols for delivered currencies or for currencies that you might add.

Country Displays the code for the country from which the currency originates.

Decimal Positions Enter the number of decimal positions that should appear in the notation for the currency. For example, use two decimal positions for Australian dollars (5.00 AUD), but no decimal positions for Japanese yen (500 JPY).

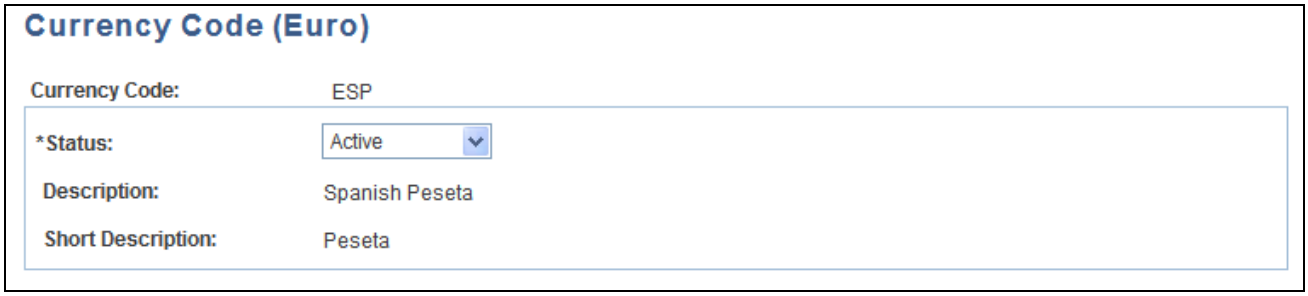
Scale Positions Enter the scale positions that you want to round for this currency. Scale positions control how many numbers appear to the left of the decimal when displayed. The data is stored with full precision in the database. For example, if you want all dollar amounts in the millions displayed as the number of millions without the zeros, enter *6* as your scale position. In this case, 24,000,000 is displayed as 24, but is stored in the database as 24,000,000.

Note. The data on this page is stored in the Currency Code table. The values on this table are effective-dated. The software is shipped with the Currency Code table in compliance with ISO standards for decimal positions. You can increase the number of decimals to a maximum of three.

Note. PeopleSoft updates the Currency Code table and the fully populated country, state, and province code tables as national boundaries and designations change.

Updating the Status of a Euro Currency Code

Access the Currency Code (Euro) page (EPM Foundation, EPM Setup, Common Definitions, Currencies and Rates, Currency Code (Euro)).



Currency Code (Euro) page

Status Displays whether a currency is an *active* participant in the euro.

Note. Do not attempt to modify the currency quotation methods for currencies that are linked to the euro.

Setting Up Market Rates for EPM Currency Conversion

The PeopleSoft approach to market rates and currency conversion is driven by the need to accommodate business practices related to the European Common Currency (euro). In addition to currency exchange rates, PeopleSoft supports the many different types of global market rates, such as interest rates, stock exchange indexes, and economic indicators.

PeopleSoft-delivered ETL jobs move market rate data from your source transaction system to EPM target warehouse tables. You need only use the market rate PIA pages to redefine or modify your existing market rate data.

This section discusses how to:

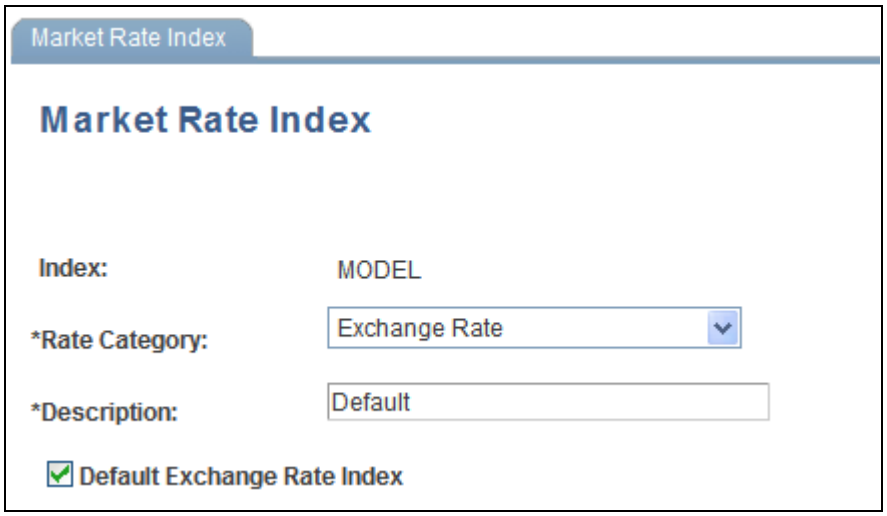
- Define market rate indexes.
- Define market rate types.
- Define market rate tolerances.

Pages Used to Manage Market Rates

Page Name	*Definition Name*	*Navigation*	*Usage*
Market Rate Index	RT_INDEX_TBL	EPM Foundation, EPM Setup, Common Definitions, Currencies and Rates, Market Rate Index	Describe the indices for which you track rates. Typical market rate indexes include LIBOR, Bloomberg foreign exchange, and Reuters foreign exchange. An index categorizes the various market rates that you track.
Market Rate Type	RT_TYPE_TBL	EPM Foundation, EPM Setup, Common Definitions, Currencies and Rates, Market Rate Type	Define market rate types. Rate types include commercial, floating, average, and historical.
Market Rate Definition	RT_RATE_DEF_TBL	EPM Foundation, EPM Setup, Common Definitions, Currencies and Rates, Market Rate Definition,	Define tolerance limits for rates and determine what action should occur if a new rate occurs outside of the tolerance limit. The fields on this page differ according to the rate category of the market rate index.

Defining Market Rate Indexes

Access the Market Rate Index page (EPM Foundation, EPM Setup, Common Definitions, Currencies and Rates, Market Rate Index).



Market Rate Index page

Rate Category Select a Rate Category from the list: Options include: *Commodity Price, Economic Indicator, Exchange Rate, Futures Price, Interest Rate, Other, Stock Exchange, Index,* or *Stock Price.*

Default Exchange Rate Index If you are entering exchange rate indexes, select this check box to indicate which index should be used to retrieve currency exchange rates. You can specify only one index code as the default.

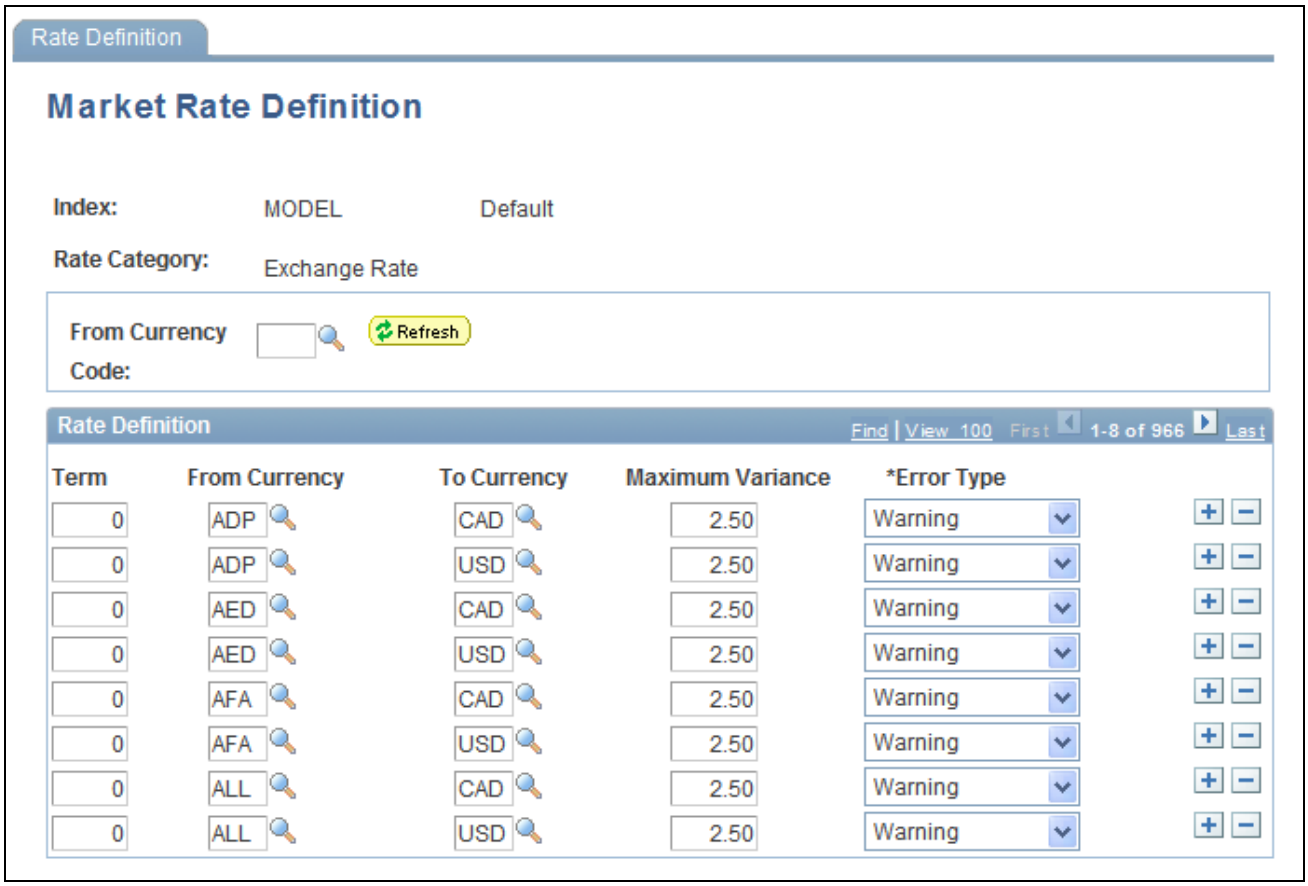
Defining Market Rate Types

Access the Market Rate Type page (EPM Foundation, EPM Setup, Common Definitions, Currencies and Rates, Market Rate Type).

Enter a description and short description for each market rate type that you use.

Defining Market Rate Tolerances

Access the Market Rate Definition page (EPM Foundation, EPM Setup, Common Definitions, Currencies and Rates, Market Rate Definition).



Market Rate Definition page

The information that appears on this page depends on the type of index that you select. For example, if you select an index with interest rates, interest rates appear on this page.

Market rate definitions specify the valid term, currency, and other appropriate field combinations for market rates. For example, if you have a market rate definition for an exchange rate with a term of *30*, a from currency of *EUR*, and a to currency of *USD*, you can enter a rate using this combination in the market rate table. If you do not have a market rate definition, the system creates one for you using the default values.

From Currency Code Enter the *from* currency code.

Refresh Click the Refresh button to limit the display to the specified *from* currency code.

Term Enter the desired term (expressed in days). A zero term indicates a spot rate.

From Currency Enter the appropriate *from* currency. This value is used with the *to* currency value as part of an exchange rate pair. When you use triangulation, include a definition for each of the currency pairs involved in the triangulation.

To Currency Enter the appropriate *to* currency. This value is used with the *from* currency value as part of an exchange rate pair.

Maximum Variance Indicate the percentage of variance that is allowed when the user maintains the market rate. The system generates an error message if the change exceeds the tolerance. The default value is *2.50* (2.5 percent).

Error Type Enter the type of error processing that should occur if the maximum variance is exceeded. Values are:

None: No error processing occurs, and the new rate is used even though it exceeds the limit.

Stop: Processing halts, and the system prevents you from saving the new rate.

Warning: Default value. A warning appears; you can ignore it and save the new rate.

Defining Currency Quotations for EPM Currency Conversion

This section provides an overview of currency quotations and discusses how to maintain currency quotation methods.

Understanding Currency Quotations

PeopleSoft supports direct and indirect rate quotation, quote units, and triangulation. These are flexible and accurate tools that enable you to convert and manage multicurrency operations.

The currency quotation method controls how a stored rate is displayed and how an entered rate is interpreted and stored in the database. You maintain a currency quotation method for each *from* currency and *to* currency pair.

You do not typically maintain rates online for currency pairs that triangulate. Instead, the Cross rate/Triangulation Generation SQR determines the cross rate by using the rates between the *from* currency and the reference currency, and between the reference currency and the *to* currency. Currency quotation methods must be set up correctly to yield the desired triangulation results.

PeopleSoft-delivered ETL jobs move currency quotation data from your source transaction system to EPM target warehouse tables. You need only use the Currency Quotation Method PIA page to redefine or modify your existing currency quotation data.

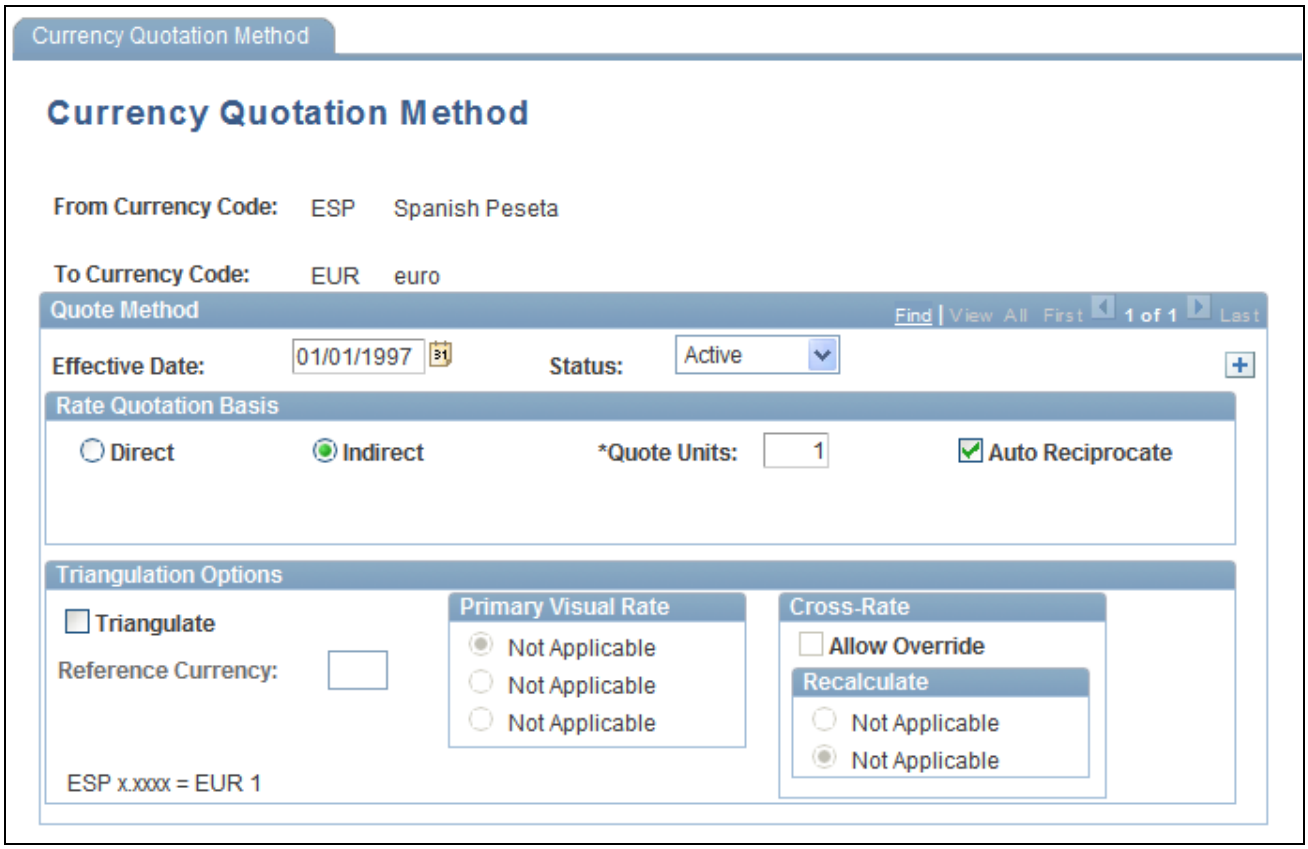
Note. Define currency quotation options before you enter and calculate the rates.

Page Used to Define and Maintain Currency Quotations

Page Name	*Definition Name*	*Navigation*	*Usage*
Currency Quotation Method	CURR_QUOTE_PNL	EPM Foundation, EPM Setup, Common Definitions, Currencies and Rates, Currency Quotation Method	Maintain a currency quotation method for each *from* currency and *to* currency pair.

Maintaining Currency Quotation Methods

Access the Currency Quotation Method page (EPM Foundation, EPM Setup, Common Definitions, Currencies and Rates, Currency Quotation Method).



Currency Quotation Method page

Rate Quotation Basis

Direct Determine whether you want the rates for a currency pair quoted directly. For example, in converting United States dollars (USD) to euro (EUR), a direct quote would indicate that USD 1 = EUR x.xxxx. In this case, euros are quoted directly against the United States dollar.

Indirect Determine whether you want the rates for a currency pair quoted indirectly. For example, in converting United States dollars (USD) to euros (EUR), an indirect quote would indicate that USD x.xxxx = EUR 1.

Quote Units Sometimes called "scaling factors," these preserve decimal position. You can enter any value in this field, although quote units generally are on a scale of 10 (such as 10, 100, 1000).

The default value for this field is *1*.

Auto Reciprocate Select this check box to create or update the rate for the reciprocal currency pair automatically whenever an exchange rate is added or updated. For example, when you enter a new USD to GBP rate, the GBP to USD rate will be updated automatically. You can only apply this feature to currency pairs for which quotation methods have been established.

Note. Currency pairs that triangulate must be classified as either direct or indirect to use in displaying the calculated cross rate. Two fields store the rate conversion factor: RATE_DIV and RATE_MULT. The currency conversion formula is always: (From currency ÷ RATE_DIV) ? (RATE_MULT) = To currency

---

Triangulation Options

Select the Triangulate check box to have the system convert two currencies through a third currency (the reference currency). Triangulation is used in hyperinflationary environments in which all conversions to the local currency are done through a more stable currency such as USD.

Note. Triangulation was initially used for European countries participating in the euro. However, since 1999 all countries participating in the euro are quoted directly against the foreign currency.

Any countries newly participating in the euro might be initially subject to triangulation, however. The triangulation example below and any other examples in this chapter that show triangulation, use a fictional country, with a currency code of NEW, that has just joined the euro. This country is subject to triangulation.

The following are examples of indirect quotation, direct quotation with quote units, and triangulation:

USD 100 to GBP (indirect) = (USD 100 x 1.6) x 1 = GBP 62.50.

CHF 1000 to German marks (DEM) (direct with units) = (CHF 1000 / 100) x 119.335 = DEM 1193.35.

USD 100 to NEW (triangulate) = (USD 100 / 1.25) x 6.8 = NEW 544.

For example, to convert from USD to NEW with triangulation, you perform two conversions:

- Convert the USD amount to the reference currency using the appropriate triangulated rate.

  The triangulated rate uses the USD to EUR component of the USD to NEW triangulated rate that is stored in RATE_DIV.

- Convert the reference currency to NEW using the fixed exchange rate.

  The exchange rate uses the EUR to NEW component of the USD to NEW triangulated rate that is stored in RATE_MULT.

Typically, you do not maintain triangulation rates directly. Instead, you process these and all rates through the Cross/Reciprocal Rate Calculator.

Select a reference currency through which the *from* currency will be converted.

You must consider three exchange rates for triangulated currency pairs:

- The rate between the *from* currency code and reference currencies.
- The rate between the reference and the *to* currency code currencies.
- The cross rate between the *from* currency code and *to* currency code currencies.

Primary Visual Rate

Select one of the three conversion rates as the primary rate that appears on primary pages and reports.

Cross Rate

Select the Allow Override check box to enable users to override the cross rate for a triangulated currency pair. If this option is deselected, you can change the components of only the triangulated rate.

Recalculate

If you select the Allow Override check box, you must maintain triangulation accuracy by specifying which currency pair the system should use to recalculate if the cross rate is overridden.

Automatic Reciprocation of Quote Methods

The Currency Quotation Method page automatically provides reciprocal methods. For example, if you define the conversion of USD to NEW as indirect, this record is automatically created to indicate a quote method of direct. If you change the quote method on the NEW to USD record, the USD to NEW record is updated automatically.

Note. This example uses a hypothetical currency NEW that has just begun participating in the euro and is still subject to triangulation for an initial period.

Using the conversion of USD to NEW as an example, this table shows each possible field value and its corresponding reciprocal value.

Field	*Value (for USD to NEW)*	*Reciprocal Value (for NEW to USD)*
Quotation Basis	Direct Indirect	Indirect Direct
Quote Units	Any valid value	Same value
Rate Decimal Positions	4 (default value)	Same value
Auto Reciprocate	Yes No	Yes No
Triangulate	Yes No	Yes No
Reference Currency	Any valid value	Same value
Primary Visual Rate	From - To (USD - NEW) From - Ref (USD - EUR) Ref - To (EUR - NEW)	From - To (NEW - USD) Ref - To (EUR - USD) From - Ref (NEW - EUR)
Cross rate Allow Override	Yes No	Yes No
Cross rate Recalculate	From - Ref (USD - EUR) Ref - To (EUR - NEW)	Ref - To (EUR - USD) From - Ref (NEW - EUR)

Establishing Market Rates for EPM Currency Conversion

PeopleSoft-delivered ETL jobs move market rate data from your source transaction system to EPM target warehouse tables. You need only use the market rate PIA pages to redefine or modify your existing market rate data.

This section discusses how to:

- Define market rates.
- Establish rate definitions.
- Maintain exchange rates.

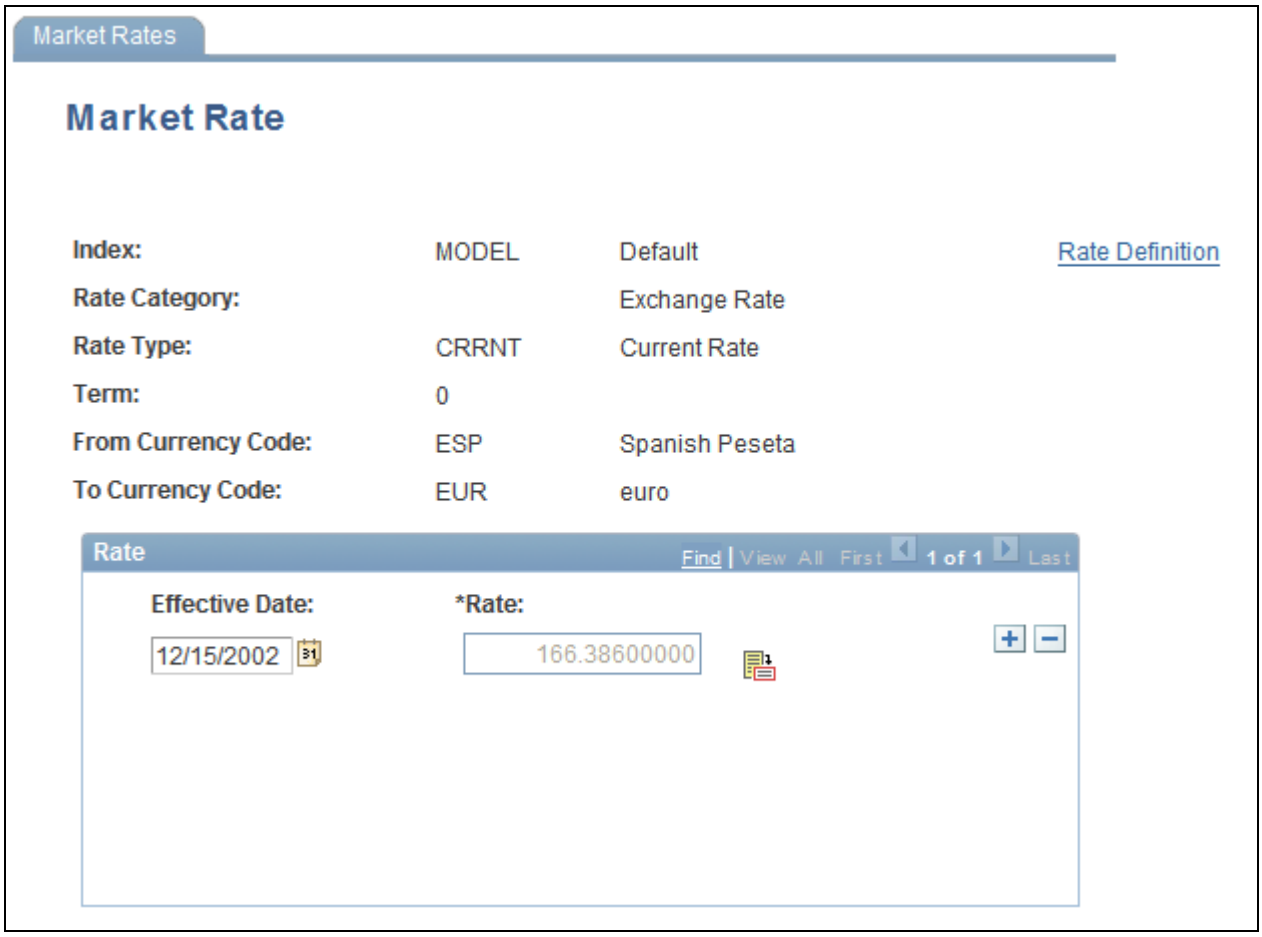
- Load market rates.

Pages Used to Establish Market Rates

Page Name	*Definition Name*	*Navigation*	*Usage*
Market Rate	RT_RATE_PNL	EPM Foundation, EPM Setup, Common Definitions, Currencies and Rates, Market Rates, Market Rate	Define market rates.
Rate Definition	RT_RATE_DEF_SEC	Click the Rate Definition link on the Market Rates page.	Establish rate definitions.
Exchange Rate Detail	EXCH_RT_DTL	Click the Exchange Rate Detail button on the Market Rates page. This page can also be accessed from other pages in the system.	Maintain exchange rates.

Defining Market Rates

Access the Market Rates page (EPM Foundation, EPM Setup, Common Definitions, Currencies and Rates, Market Rates, Market Rate).



Market Rates page

Rate The Rate field displays the *visual rate*or, in the case of triangulated exchange rates, the *primary visual rate.* The visual rate is typically the cross rate, but it can also be one of the other component rates of the triangle.

You can edit the rate for non triangulated rates and for triangulated rates if a quotation method has been defined for the currency pair and the Cross rate Allow Overridecheck box is selected on the Currency Quotation Method page. If an override is not allowed, you can update the exchange rate values to and from the reference currency on the Exchange Rate Detail page.

Changing a Triangulated Cross Rate

If you change a triangulated cross rate, the system recalculates one of the component rates. This can result in the cross rate being recalculated in a manner that is slightly different from the one that you entered. For example, you start with a triangulated rate of RM=6.80000000 and RD=1.25000000 for a cross rate of 5.44000000. If you change the cross rate to 5.43550000:

- The system first recalculates RD to 1.25103486.
- The system then recalculates the cross rate to 5.43550001 based on the first recalculation.

In rate maintenance, you must accept the recalculation. However, in a situation such as journal entry, a warning message enables you to override triangulation and to use the exact rate that you entered, which results in the rate being stored as RM equal to 5.43550000 and RD equal to 1.

You can edit the Rate field except when *all* of these conditions are true:

- The rate is triangulated.
- The primary visual rate is the cross rate.
- The Allow Override check box on the Currency Quotation Method page is deselected for the exchange rate's quotation method.

Note. Typically, you do not maintain triangulated exchange rates online. Instead, maintain the rates of the *from* currency to the *reference* currency and the *reference* currency to the *to* currency, and then run the Cross rate Reciprocal SQR (EO9030.SQR) to define the triangulated exchange rates.

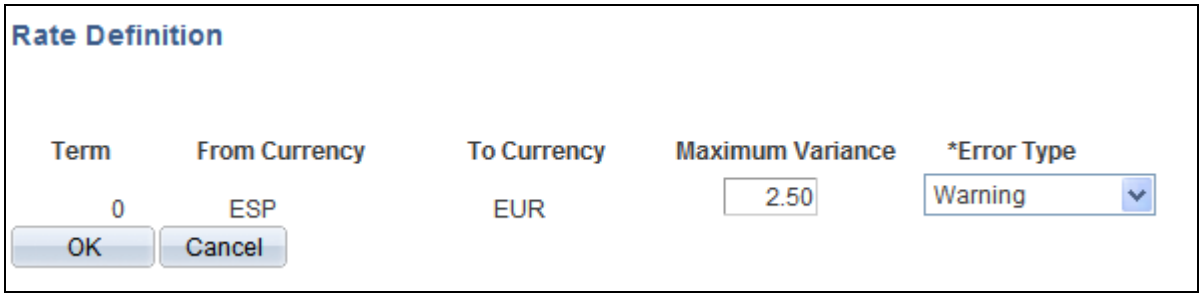
Reciprocal Currency Pairs

If a quotation method has been defined for the currency pair and if the Auto Reciprocate check box is selected, creating or maintaining a rate for a currency pair automatically creates or updates the rate of the reciprocal currency pair. For example, if you change the USD to GBP rate, the GBP to USD rate automatically is automatically updated. You can only automatically reciprocate currency pairs for which quotation methods have been defined.

If a rate definition does not already exist for the currency pair, one will be automatically created with the default values of 2.5 percent maximum variance and warning message processing.

Establishing Rate Definitions

Access the Market Rate - Rate Definition page (Click the Rate Definition link on the Market Rates page).



Market Rate - Rate Definition page

Maximum Variance You can modify the maximum variance—that is, the percentage of variance allowed when you maintain the market rate. If the change exceeds the tolerance, an error results. The default value is *2.50* (2.5 percent).

Error Type You can modify the type of error that results when the tolerance defined in the Maximum Variance field is exceeded during data entry. Error type values are:

None: No error processing occurs. The new rate is used even though it exceeds the limit.

Stop : Processing halts. The system prevents you from saving the new rate.

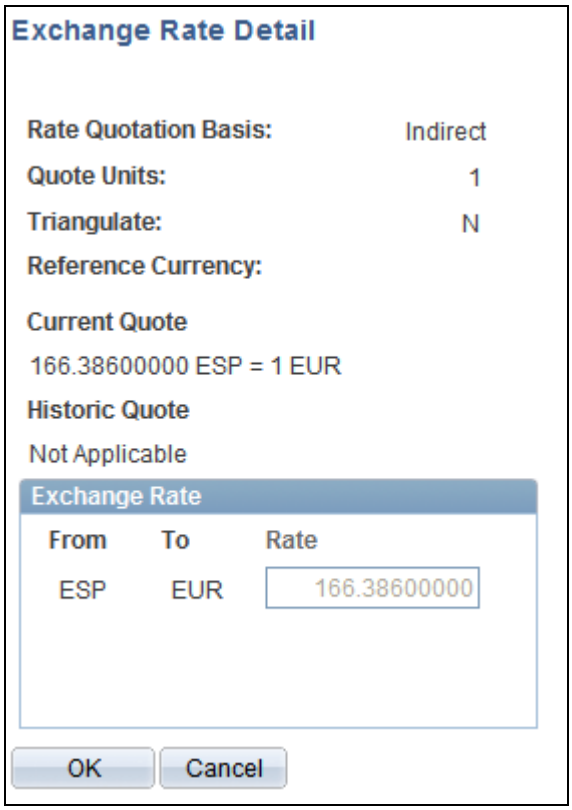
Warning: A warning appears. You can ignore it and save the new rate.

Note. The results of changing the rate definition do not take effect until you save the Market Rates page.

Click OK to return to the Market Rates page.

Maintaining Exchange Rates

Access the Market Rate - Exchange Rate Detail page (Click the Exchange Rate Detail button on the Market Rates page. This page can also be accessed from other pages in the system.).



Market Rate - Exchange Rate Detail page

The read-only fields include:

Rate Quotation Basis Displays the quotation basis for the exchange rate as it is defined on the Currency Quotation Method page. If no quotation method is defined, the quotation basis is *Direct.*

Quote Units Displays the quote units for the exchange rate as defined on the Currency Quotation Method page. If no quotation method is defined, the quote unit is *1*.

Triangulate Displays the triangulated setting for the exchange rate as it is defined on the Currency Quotation Method page. If no quotation method is defined, the triangulated setting is *N*.

Reference Currency Displays triangulated exchange rates only, and shows the reference currency used in the triangulated exchange.

Current Quote Displays the current exchange rate used to convert the from currency to the to currency. A direct, non-triangulated rate shows quote units (or 1) on the left side of the equal sign and the visual rate on the right. For example: 1 USD = 1.40000000 CAD.

A triangulated rate displays two component rates of the triangle: the rate for converting the *from* currency to the *reference* currency, and the rate for converting the *reference* currency to the *to* currency.

Historic Quote Displays a quote to indicate the quotation method originally used by a historic exchange rate if the system determines that the original quotation method of the historic rate differs from the current quotation method. This field displays the following information:

- A quote, if the historic rate has converted the *from* currency to the *to* currency directly using a calculated reciprocal rate, but the current quotation method for the currency pair is now indirect.
- A quote, if the historic quote method were non-triangulated and the current quote method is triangulated.
- *Not Applicable*, if the system does not determine that the historic and current quote methods are different.

Exchange Rate Displays a single visual rate for non-triangulated exchange rates or displays all three component visual rates for triangulated exchange rates. The cross rate for triangulated exchange rates is editable only if the Allow Override check box is selected in the exchange rate's quotation method definition.

Loading Market Rates

Market rates can be loaded to the RT_RATE_TBL from any external source using the DataStage ETL tool.

Note. Use the Market Rates page to verify that the market rates were loaded correctly.

Calculating Currency Rates for EPM Currency Conversion

PeopleSoft-delivered ETL jobs move currency rate data from your source transaction system to EPM target warehouse tables. You need only use the currency rate PIA pages to redefine or modify your existing currency rate data.

This section provide an overview of currency calculations and discusses how to:

- Calculate cross rates and reciprocal rates.
- Run the Currency Exchange Calculator tool.

Understanding Currency Calculations

PeopleSoft calculates currency rates for cross rates, triangulated rates, and reciprocal rates.

EPM utilizes two tools for currency calculations:

- The Currency Exchange Calculator tool quickly performs *ad hoc* currency conversion using the exchange rates that are stored on the market rates table.
- The Cross/Reciprocal Rate Calculator calculates exchange rates and updates the market rates table.

  It performs three functions by generating the rates shown in this table:

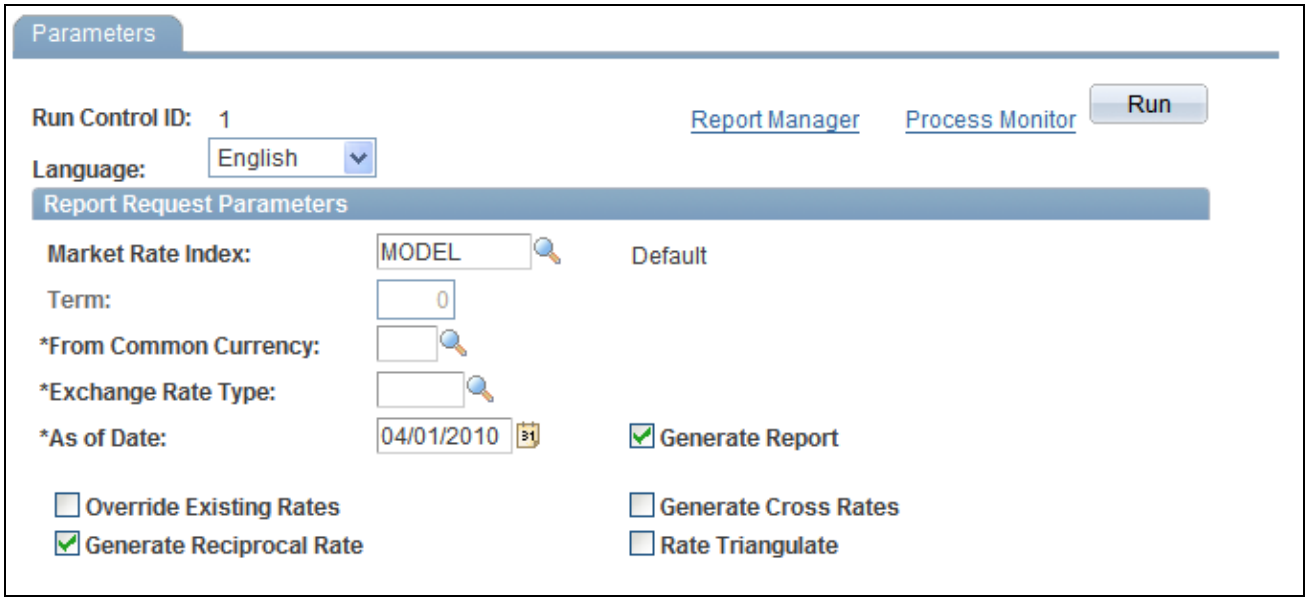
Cross rates for nontriangulated currency pairs	For example, an organization subscribes to a rate service that provides all rates respective to the USD. Starting with a USD to Canadian dollar rate and a USD to Mexican peso rate, the system can calculate a new Canadian dollar to Mexican peso cross rate.
Triangulated rates for triangulated currency pairs	For example, the euro to NEW (a fictitious country that has just joined the euro and is subject to triangulation) fixed rate has been established on the market rate table and a new euro to USD rate has just been entered. Using this information, the process can create a new USD to NEW triangulated rate. The difference between triangulated rates and cross rates affects how the data is stored in the database. When calculating a cross rate, you actually create a new rate. When calculating a triangulated rate, the individual components of the source rates are stored on the target.
Reciprocal rates for those currency pairs that are not automatically reciprocated	For example, using a USD to CAD rate as the source, the process calculates the CAD to USD reciprocal. If quote methods are in place, the visual rate remains the same and a difference exists in how the data is stored in the database (RATE_MULT and RATE_DIV are inverse). If quote methods are not used, the process calculates an inverse rate, meaning that the visual rates differ.

Pages Used to Calculate Currency Rates

Page Name	*Definition Name*	*Navigation*	*Usage*
Cross/Reciprocal Rate Calc (calculator)	RUN_FIN9030	EPM Foundation, EPM Setup, Common Definitions, Currencies and Rates, Cross/Reciprocal Rate Calc, Cross/Reciprocal Rate Calculator, Cross/Reciprocal Rate Parameters	Calculate cross rates and reciprocal rates by defining parameters to run the FIN9030 SQR report.
Currency Exchange Calculator	CURRENCY_EXCHNG_PN	EPM Foundation, EPM Setup, Common Definitions, Currencies and Rates, Currency Exchange Calculator	Run the Currency Exchange Calculator to calculate the currency exchange between currencies.

Calculating Cross Rates and Reciprocal Rates

Access the Cross/Reciprocal Rate Calculator page (EPM Foundation, EPM Setup, Common Definitions, Currencies and Rates, Cross/Reciprocal Rate Calc, Cross/Reciprocal Rate Calculator, Cross/Reciprocal Rate Parameters page).



Cross/Reciprocal Rate Parameters page

Language Code Select the language for translation.

Market Rate Index Select a market rate index.

Term	The system displays this value by default from the value entered on the Market Rate Definition page.
From Common Currency	Select a currency code from which to calculate a reciprocal rate.
Exchange Rate Type	Select the type of exchange rate to use for this calculation.
As of Date	Determines the effective date of newly created exchange rates (the output of the process) and rates that are used as the basis for calculations (the input to the process). The report uses the most current currency quotation method for the currency pair as input for the process. If the as of date is the current effective rate as of the specified date, it can affect triangulation.
Generate Report	Select to specify that the system generate a report that displays exchange rates and reciprocal and cross rate calculations.
Override Existing Rates	Select to direct the calculated rate to override rates for the exchange rate type, regardless of the as of date.

Generating Reciprocal Rates

You can select the Generate Reciprocal Rate check box by itself or in combination with the Rate Triangulate and Generate Cross Rate check boxes. When this option is selected, the system automatically calculates reciprocal rates for currency pairs for which the autoreciprocate option on the currency method is disabled.

The system does not directly manipulate the exchange rates. The system uses numerator and denominator values such that:

(From currency / RATE_DIV) x RATE_MULT = To currency

Generating Cross Rates

When you select the Generate Cross Rates check box, the system automatically generate cross rates.For example, to generate cross-currency rates for USD, (CAD), and (MXP), you enter USD to CAD = 1.473 and USD to MXP = 9.8793. The system automatically generates CAD to MXP = 9.8793 / 1.473 = 6.7069246.

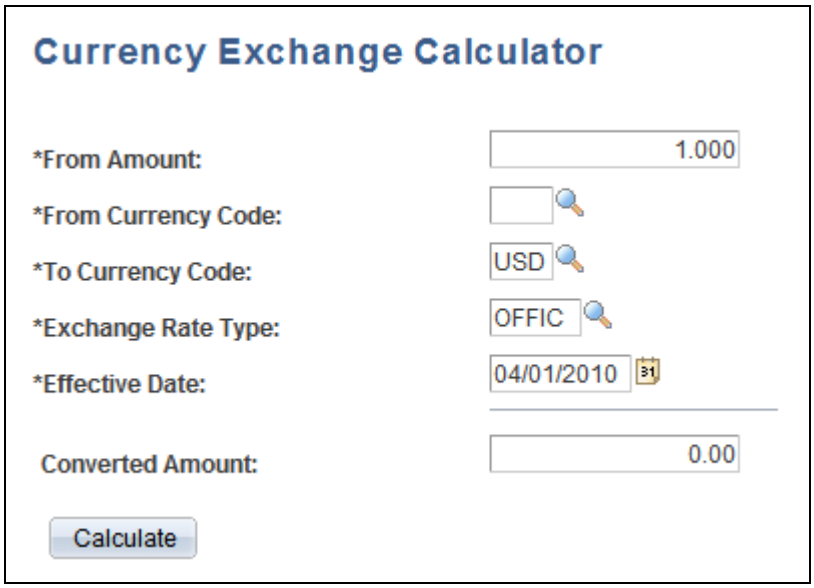
If you generate cross rates, you must select a *from* currency and a *to* currency. You can enter a wild card of *%* in either field or both fields to indicate*from all* or*to all* currencies.

Generating Triangulation Rates

When you select the Rate Triangulate check box, the system converts two currencies through a third currency. If you select rate triangulation, you must select a *from* currency and a *to* currency. You can enter a wild card of *%* in either field or both fields to indicate *from all* or *to all* currencies.

Running the Currency Exchange Calculator Tool

Access the Currency Exchange Calculator page (EPM Foundation, EPM Setup, Common Definitions, Currencies and Rates, Currency Exchange Calculator).



Currency Exchange Calculator page

This tool enables you to enter a rate or an amount in a currency other than the base currency, or to compute an exchange using an alternative rate type.

From Amount Enter a *from* amount. The currency exchange is based on the *from* amount that you enter and the current exchange rate set up on the Market Rates page.

From Currency Code Select the currency code from which to calculate the exchange amount.

To Currency Code Select the currency code to which to calculate the exchange amount.

Exchange Rate Type Select the type of exchange rate to use for this calculation.

Converted Amount Displays the converted amount. The system automatically calculates this amount when you save the page.

Note. Do not decrease the number of decimals after you have created transactions for that currency; the system will not properly round the previous rounded amount fields with the new precision.

Configuring Currency Precision for Currency Conversion

This section provides an overview of currency precision and discusses how to:

- Activate currency precision.
- Maintain currency precision by currency.
- Report with currency precision.

Understanding Currency Precision

According to the ISO standard, currency precision can range from zero decimals to three decimals. For example, USD amounts have two digits to the right of the decimal, and JPY have none. To support this dynamic currency precision, the system delivers all of its currency-sensitive amount fields with a standard length of 23.3, or 23 digits to the left of the decimal and three digits to the right. A control currency on the same record exists to control the display and processing of such amount fields.

PeopleSoft applications round all currency-sensitive amount fields to the currency precision of the controlled currency during all online or background processes. For example, in a database that contains amount fields with a length of 23.3, JPY are rounded to 123.000 and USD are rounded to 123.230. The system does not place a nonzero after the decimal for a JPY amount or after the second digit to the right of the decimal for a USD amount.

Although amount fields are stored in the database with decimal placeholders, the system displays amount fields with the precision that is appropriate for the currency. For example, it displays JPY as 123 and USD as 123.23. When you enter an amount, you cannot enter more than the defined precision. If you attempt to do so, the system treats the entry as an online error.

PeopleCode programs and background processes round all currency-sensitive amount fields to the currency precision of the controlled currency.

PeopleSoft-delivered ETL jobs move currency precision data from your source transaction system to EPM target warehouse tables. You need only use the currency precision PIA pages to redefine or modify your existing currency precision data.

See Also

Chapter 5, "Setting Up Currency Rules for EPM," Configuring Currency Precision for Currency Conversion, page 114

Activating Currency Precision

Currency precision is a PeopleSoft PeopleTools option. When it is selected using the PeopleTools Options page, all features of currency precision are activated. When the option is deselected, all amount fields behave as if no controlled currency exists. The system displays amount fields as defined in the PeopleSoft Application Designer and rounds them to the number of decimals defined in the Application Designer.

Note. If you deselect the multicurrency check box, the system only supports the default amount field size of 15.3—it does not support the larger amount field size of 23.3. After you deselect this check box, selecting it again does not automatically round existing transaction amounts.

Maintaining Currency Precision by Currency

Use the Currency Code page to access the currency code table, in which you define the decimal position by currency. The values in this table are effective dated. The software is shipped with the currency code table in compliance with ISO standards for decimal positions. You can increase the number of decimals to a maximum of three.

Warning! Do *not* decrease the number of decimals after transactions are entered in that currency; the system does not properly round the previously rounded amount fields with the new precision.

See Also

Chapter 5, "Setting Up Currency Rules for EPM," Maintaining Currency Codes, page 96

Reporting with Currency Precision

Most PeopleSoft SQR reports display currency-controlled amounts with the number of decimal places that are defined by the associated currency. For example, a JPY amount appears as 123 on a report, and a United USD amount appears as 123.23.

Amounts on Crystal and PS/nVision (Microsoft Excel) reports appear as two-decimal-place numbers. If you want to show three decimal places on these reports, you must configure the reports to do so.

Third-party reporting tools used by PeopleSoft do not fully support numeric fields greater than 15 digits. Microsoft Excel uses an eight-byte float for numeric fields, which causes values to be truncated after the fifteenth digit.

Crystal displays up to 15 digits correctly. When a value exceeds 15 digits, Crystal inserts invalid numbers into the decimal positions. This is an issue for only very large currency amounts. For any of these reporting tools, the accuracy of the results is:

- Hundreds of trillions of yen (precision = 0).
- Trillions of dollars (precision = 2).
- Hundreds of billions of dinar (precision = 3).

For example, if you populate a 23.3 numeric database amount field with the number *2*, the following table illustrates the number that is displayed in each type of report.

Number of Digits	*Crystal*	*Excel*	*SQR*
16	2,222,222,222,222.222	2,222,222,222,222.220	2,222,222,222,222.222
17	22,222,222,222,222.219	22,222,222,222,222.200	22,222,222,222,222.220
18	222,222,222,222,222.188	222,222,222,222,222.000	222,222,222,222,222.200

Chapter 6

Setting Up EPM Security

This chapter provides an overview of EPM security and setup, and discusses how to:

- Specify field level security options.
- Establish ledger security.
- Specify row and column level security.
- Establish security bridges for your business intelligence tool.

Understanding EPM Security and Setups

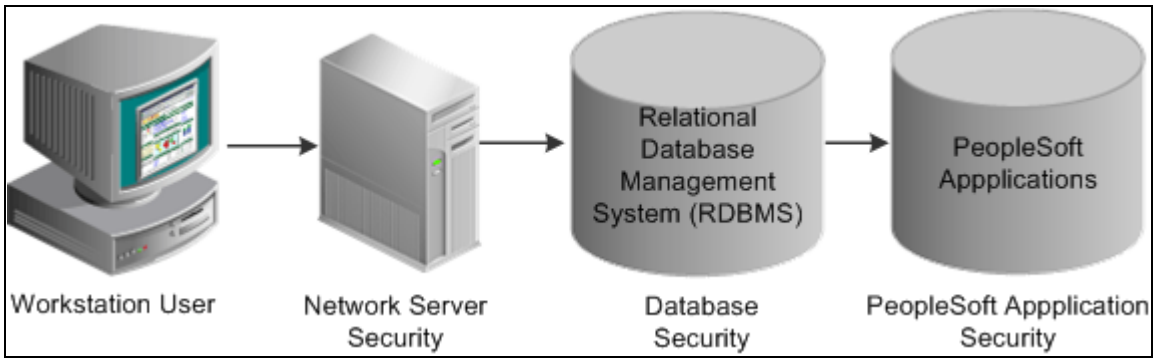
This section discusses:

- PeopleSoft application security
- EPM security
- Setting up EPM security
- EPM security views

PeopleSoft Application Security

PeopleSoft applications use multilevel security to enable you to successfully manage shared data environments. You set up data access at different entry points within your system and define the most efficient path to data across business groups, tables, departments, pages, and so forth. You have full control over security definitions, selecting options to create a matrix that enables or restricts user access to data through a series of authorizations.

Security access covers three areas: networks, databases, and applications. Network security controls the overall point of entry into your system hardware and software resources. Database security narrows the scope of a user's information access. At the application level, security extends to the field level. This diagram illustrates PeopleSoft application security levels:



Levels of security in PeopleSoft applications

Application Security Terms

This table describes the various types of PeopleSoft application security:

Security Type	*Location*	*Function*
Network	Network software	Controls entry into the network and authorizes rights to use shared resources.
Relational Database Management System (RDBMS)	Operating system	Controls access to the database.
User	PeopleTools	Controls access to application pages, functions, and business components.
Object	PeopleTools	Controls access to objects or object groups that are used in application development.
Query	PeopleTools	Defines table row sets that are accessed for performing system queries.
Row-level	PeopleTools and PeopleSoft applications	Controls access to the subset of data rows within tables that the user is authorized to review or update.
Field-level	PeopleCode	Controls access to individual fields on pages.

EPM Security

EPM security controls access to specific data within the EPM database and enables you to grant users access to specific rows, columns, or fields in an EPM database table. Data-access privileges within the warehouse can be defined for both Operational Warehouse - Enriched (OWE) and Multidimensional Warehouse (MDW) tables, and for dimension and fact tables.

You can specify EPM security access using any of the following methods:

- Field-level security
- Ledger security
- Dimension (row-level) security
- Metric (column-level) security

EPM security provides a single point of entry for defining and maintaining data access rules across all the EPM Warehouses and Analytical Applications.

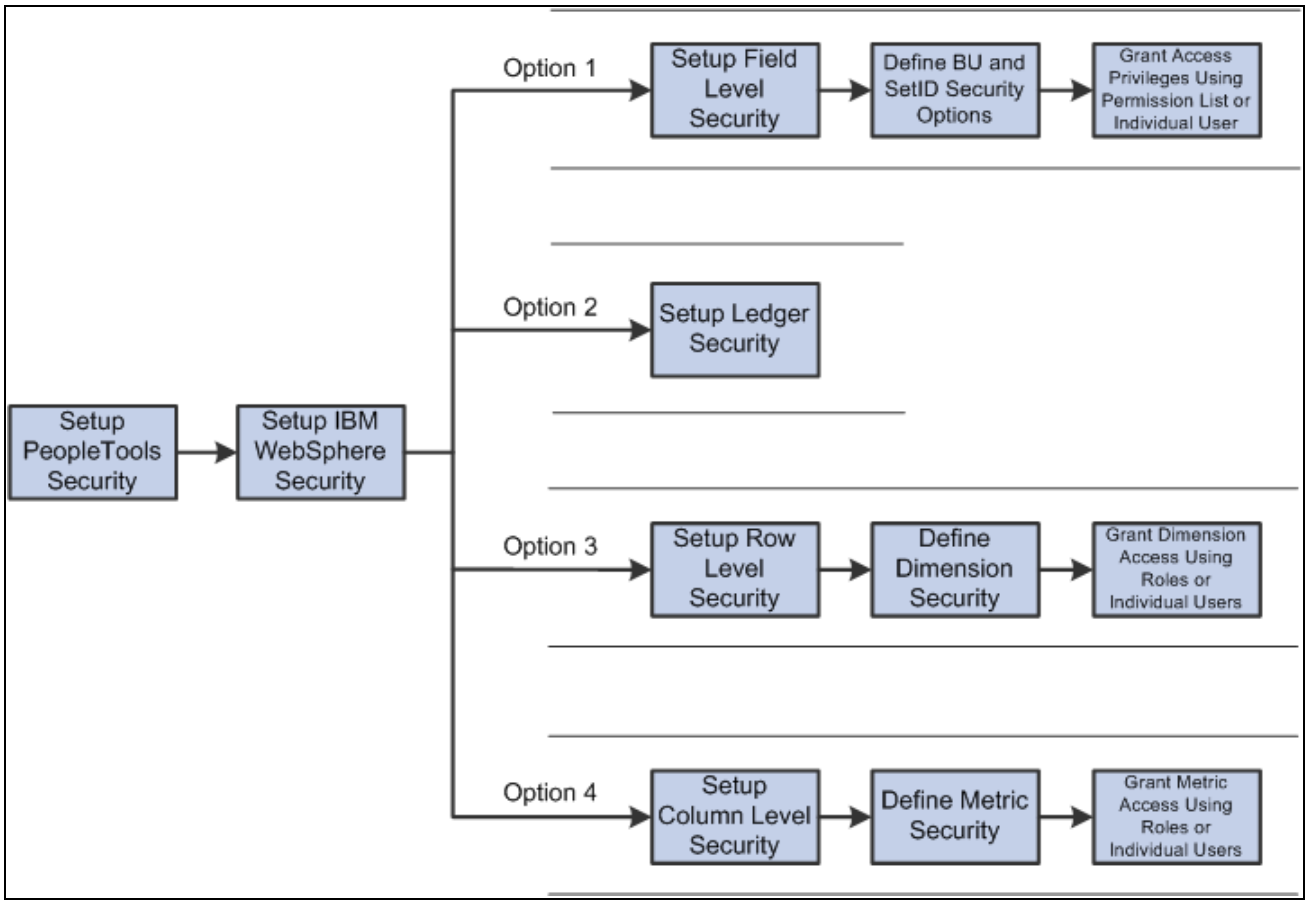
Setting Up EPM Security

EPM security is extremely flexible and enables you to restrict user access to EPM database tables in a variety of ways, including field level, row level, and column level restrictions. PeopleSoft-delivered ETL jobs move security data from your source transaction system to EPM target warehouse tables. You must set up PeopleTools and IBM WebSphere DataStage security, but you need only use the security pages documented in this chapter to redefine or modify your existing security data.

You have the option of implementing only one type of EPM security restriction or all four types:

- Field-level security.
- Ledger security.
- Row-level security.
- Column-level security.

The following security setup process flow demonstrates the different security setup options:



EPM Security Setup Process Flow

As demonstrated in the diagram, you can implement any combination of field level, ledger, row level, and column level security restrictions.

Note. Refer to your EPM Warehouse and Analytical Application PeopleBooks for any special security setups required for those products.

PeopleTools security

PeopleTools security provides user authentication and application-level security. Users are authorized to access the EPM database using Peopletools login security. Users may be granted or denied access to each application and components therein. EPM security works together with PeopleTools security, so you must properly define PeopleTools security prior to setting up EPM security.

See *PeopleSoft Enterprise PeopleTools PeopleBook: Security Administration*

IBM WebSphere DataStage Security

PeopleSoft EPM delivers the IBM WebSphere extract, transform, and load (ETL) tool to load your source data into the Operational Warehouse - Staging (OWS) and migrate the data to the OWE and the MDW. The tool builds three layers of security into the repository and its objects. Access is granted through groups and users.

The three layers of security are:

1. A user role that is created by the EPM Production Manager.

   The Production Manager assigns Windows user groups to one of four DataStage user categories. For example, if you belong to the Windows Administrators user group, you might be assigned to the DataStage Production Manager category and have full access to all areas of a DataStage project. This is the default for DataStage users.

2. Project privileges that range from browse-only to full Production Manager privileges.

3. Object-level safeguards that prevent users from jointly accessing or overwriting the same object.

   For example, if a user is working on a job, that object is locked so that no other user can access it. Locks are automatically created to avoid repository contamination.

The following DataStage user categories are defined in PeopleSoft EPM: the EPM Production Manager, Developer, Operator, and None. The Production Manager has full access to all areas of a DataStage project, including repository objects, and can create and manipulate protected projects. The Production Manager grants access to all other groups and users. Developers have full access to all areas of a DataStage project and can create and modify ETL jobs. This is the default setting. Operators have permission to run and manage DataStage jobs. Users who are assigned to None do not have permission to sign in to DataStage.

See *IBM Information Server: Administration Guide*

Field Level Security

Field level security enables you to restrict users to specific fields on your EPM database tables based on business units and setIDs (because the tables are keyed by setID or business unit).

To set up field level security you must:

1. Enable field level security and specify related options.

2. Associate security views with specific users, permission lists, business units, or setIDs.

3. Apply the aforementioned field level security parameters.

4. Associate specific users or permission lists with the business unit and/or setID restrictions (data group restrictions) you defined in the previous steps.

See Chapter 6, "Setting Up EPM Security," Specifying Field-Level Security Options, page 123.

Ledger Security

Ledger security enables you to restrict users to specific EPM ledger tables. You can restrict access to the tables by user or permission list.

See Chapter 6, "Setting Up EPM Security," Establishing Ledger Security, page 127.

Dimension (Row Level) and Metric (Column Level) Security

Row level security enables you to secure individual rows of an EPM dimension table. Column level security enables you to secure individual columns of an EPM database table. Usually, these are measures on fact tables, but may also be attributes on a dimension table that contain sensitive data, such as Employees' Salary or the Social Security Number columns. With row level and metric level support, you can restrict access by individual user or security roles.

To set up row level security you must:

1. Define dimension security.
2. Associate the dimension with a security join table.

   Security join tables are EPM database tables that store the security profiles for users along with the corresponding dimension values for which they have access.
3. Define users and security roles.
4. Assign dimension rows to the user or security role.

To set up column level security you must:

1. Define metric security.
2. Define users and security roles.
3. Assign metric to the user or security role.

See Chapter 6, "Setting Up EPM Security," Defining Dimension and Metric Security, page 128.

See Chapter 6, "Setting Up EPM Security," Understanding Role and User Based Security for Dimensions and Metrics, page 128.

EPM Security Views

Security views are SQL SELECT statements that filter out data rows whose key values are not needed as valid access parameters. The result is that users who are authorized to access setIDs or business units see only a subset of values from these edit table values. PeopleSoft EPM delivers prepackaged security views for most securable objects in EPM and the views tell the system which table views to reference. To use these security views in PeopleTools, use the views as a search view.

Object	*Delivered EPM Security View*
Personal Data	PERSONAL_SRCH
Job Data	JOB_SRCH
Job Code	JOBCODE_SRCH
Location	LOCATION_SRCH
Scenario	SCENARIO_SRCH

You can alter these prepackaged views or build your own.

Specifying Field-Level Security Options

You can set up field-level security for business units and setIDs because EPM warehouse tables are keyed by either setID or business unit. The Security Options page enables you to enable this type of security. The page updates one row in the INSTALLATION_FS table. Only one row is ever in this table (only one type of security can be active at a time). For example, if you select operator security, only selected fields (business unit, setID, or both) have security applied.

After specifying field-level security options you must associate security views with specific users, permission lists, business units, or setIDs and run the apply security application engine process (which reads the INSTALLATION_FS table, determines the security type, such as *none*, *permission list* or *operator*, and applies your selected field-level security parameters).

After running the apply security process, you must associate specific users or permission lists with the business unit and/or setID restrictions (data group restrictions) you defined in the previous steps.

This section discusses how to:

- Set up field-level security options.
- Set up security view names.
- Establish setID security by permission list or user ID.
- Establish business unit security by permission list or user ID.

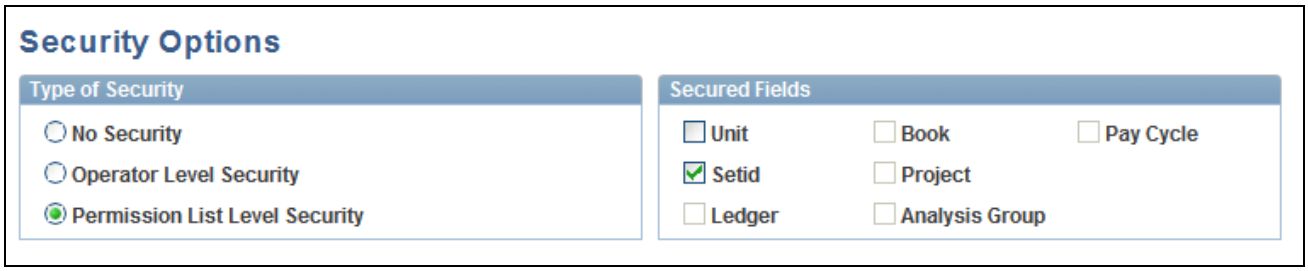
Pages Used to Specify Field-Level Security Options

Page Name	*Definition Name*	*Navigation*	*Usage*
Security Options	SECURITY_OPTIONS	EPM Foundation, EPM Security, Advanced, Security Options	Set up field-level security for business units and setIDs.
Security View Names	SECURITY_VIEWS	EPM Foundation, EPM Security, Advanced, Security View Names	Set up security view names.
Request Security Processing	RUN_FIN9001	EPM Foundation, EPM Security, Advanced, Apply Security Setups, Request Security Processing	Run the Apply Security process to apply the security setup throughout the system.
TableSet Security by Permission List	SEC_SETID_CLS	EPM Foundation, EPM Security, Advanced, TableSet Security by Perm (Permission) List	Establish setID security by permission list. Valid values for permission list are taken from your PeopleTools security setup.

Page Name	Definition Name	Navigation	Usage
TableSet Security by User ID	SEC_SETID_OPR	EPM Foundation, EPM Security, Advanced, TableSet Security by User ID	Establish setID security by user ID. Valid values for user are taken from your PeopleTools security setup.
Unit Security by Permission List	SEC_BU_CLS	EPM Foundation, EPM Security, Advanced, Unit Security by Perm (Permission) List	Establish business unit security by permission list.
Unit Security by User ID	SEC_BU_OPR	EPM Foundation, EPM Security, Advanced, Unit Security by User ID	Establish business unit security by user ID.

Setting Up Field-Level Security Options

Access the Security Options page (EPM Foundation, EPM Security, Advanced, Security Options).



Security Options page

You can implement security using any of the following fields:

- Unit: Business unit, the primary key for all transaction data.
- SetID: The primary key for all accounting structure and rules tables.
- Ledger
- Book
- Project
- Analysis Group
- Pay Cycle

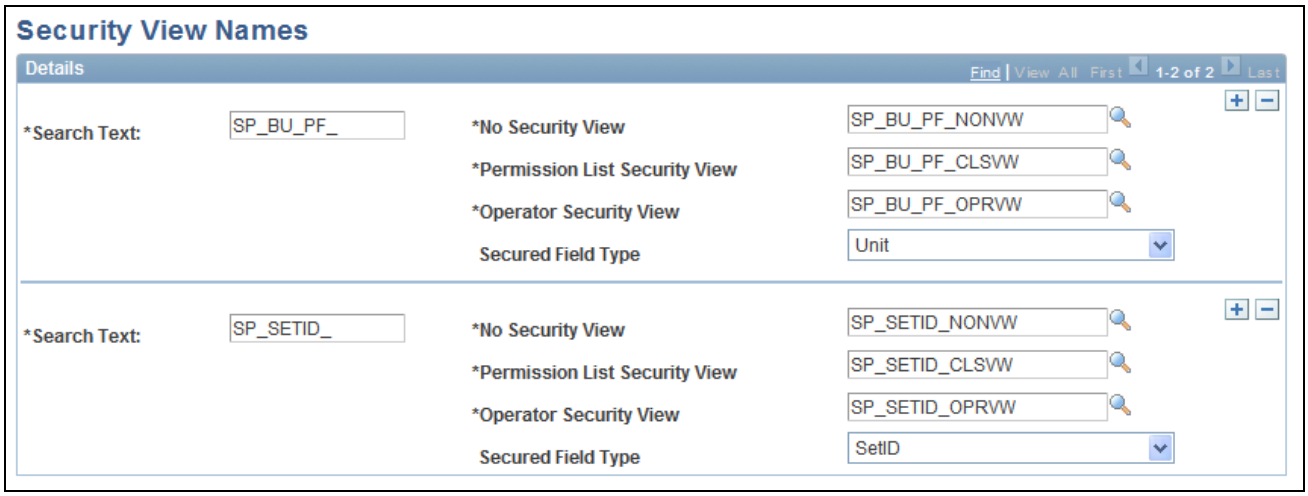
For either field, you can set the security in the following ways:

No Security Disables PeopleSoft application security. All users who are authorized to access a page may select any valid setID or business unit.

Operator Level Security	Enables PeopleSoft application security. Users are limited to the setIDs and business units that are specified by their user IDs.
Permission List Level Security	Enables PeopleSoft application security. Users are limited to the key fields that are specified by the permission list that is assigned to their user IDs. All users in a permission list have the same level of security.

Specifying Security View Names Access Privileges

Access the Security View Names page (EPM Foundation, EPM Security, Advanced, Security View Names).



Security View Names page

After you have set up your security views, you can specify which users or permission lists can access the pages that contain secured field values using the Security View Names page. Within each page, you also can hide specific fields from particular permission lists. The Security View Names page also indicates the type of field (business unit or setID) secured by each view.

The system stores the list in the SEC_VIEW_NAMES table, where you can review or update this information. You can add any security views that you configure for your system.

View names use one of three file extensions to reflect the type of security that you selected when you specified your security options:

- _NONVW: Indicates that no security has been selected for that view.
- _OPRVW: Indicates that operator (user) security has been selected for that view.
- _CLSVW: Indicates that permission list security has been selected for that view.

Use the Search Text field to search for a security view table. This field lists the view name prefixes that are supplied by each of your applications. When you run the Apply Security Setup process, the process uses these prefixes to search the system for view names that begin with these prefixes. If a view name begins with a prefix from this list, the process changes the view name extension to match the security type that you selected when you specified security options. The type of field (business unit or setID) that is secured by each view is embedded in the view name.

After you have selected your security options and defined your security views, you must run the Apply Security FIN9001 SQR process for the options to take effect.

See Also

Chapter 6, "Setting Up EPM Security," EPM Security Views, page 122

Running the Apply Security Process

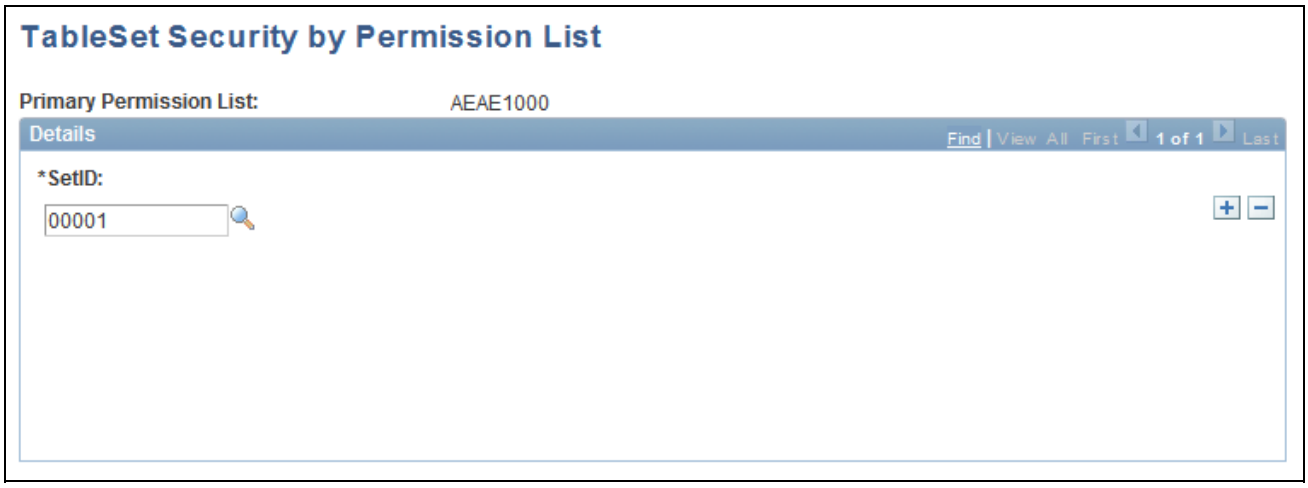
Access the Request Security Processing page (EPM Foundation, EPM Security, Advanced, Apply Security Setups, Request Security Processing).

Run the Apply Security process to apply the security parameters you set up on the Security Options and Security View Names pages. The Apply Security process reads the INSTALLATION_FS table to determine the security type (none, permission list, operator) and then switches the prompt table views to the new view names, based on SEC_VIEW_NAMES. The process searches for the values in the Search Text field on the Security View Names page and replaces the table names as specified.

Note. This process should not be confused with the request security processing process (PF_SECURITY) which is discussed later in this chapter. The PF_SECURITY process applies security parameters specified for security roles and users.

Establishing SetID Security by Permission List or User ID

Access the TableSet Security by Permission List page (EPM Foundation, EPM Security, Advanced, TableSet Security by Perm (Permission) List) or the TableSet Security by User ID page (EPM Foundation, EPM Security, Advanced, TableSet Security by User ID).

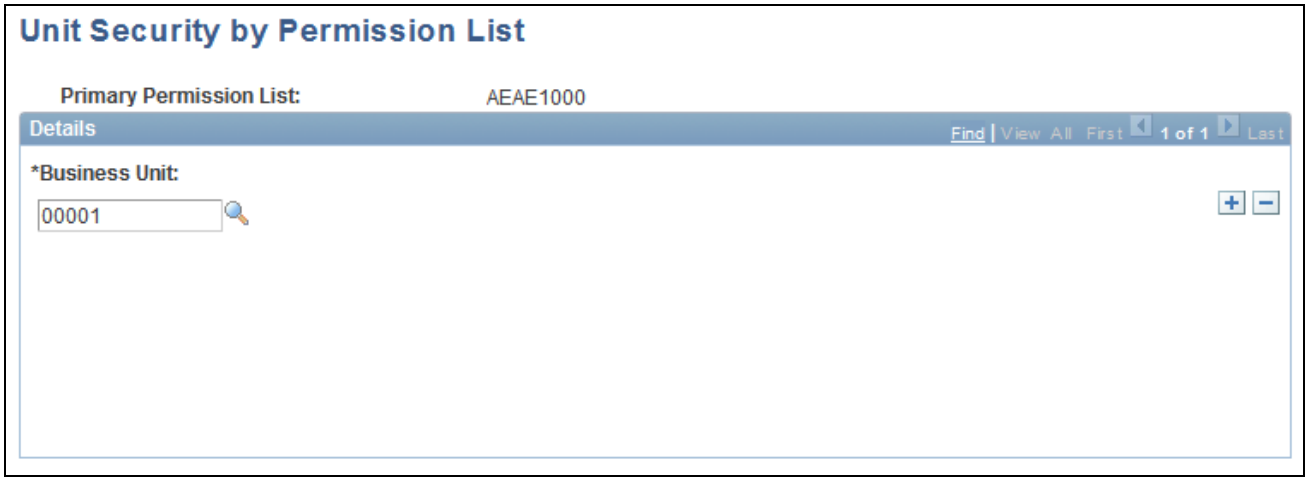


TableSet Security by Permission List page

For each user ID or primary permission list, select the setID or setIDs that the users or permission list can access.

Establishing Business Unit Security by Permission List or User ID

Access the Unit Security by Permission List page (EPM Foundation, EPM Security, Advanced, Unit Security by Perm (Permission) List) or the Unit Security by User ID page.



Unit Security by Permission List page

For each user ID or primary permission list, select the warehouse business unit or units that the user or permission list can access.

Establishing Ledger Security

You can determine which users have access to ledger data by securing access to specific ledgers. For a specified ledger you can grant access to users by permission list or user ID.

Pages Used to Define Ledger Security

Page Name	*Definition Name*	*Navigation*	*Usage*
Ledger/Perm List	SEC_LEDGER_CLS	EPM Foundation, EPM Security, Advanced, Ledger Security by Permission List, Ledger/Perm List	Select a ledger to associate with a security permission list.
Ledger by User ID	SEC_LEDGER_OPR	EPM Foundation, EPM Security, Advanced, Ledger Security by User ID, Ledger by User ID	Select a ledger to associate with a user ID.

Defining Ledger Security by Permission List or User ID

Access the Ledger/Perm List (EPM Foundation, EPM Security, Advanced, Ledger Security by Permission List, Ledger/Perm List) or Ledger by User ID page (EPM Foundation, EPM Security, Advanced, Ledger Security by User ID, Ledger by User ID).

Use these pages to select a ledger to associate with a security permission list or user ID.

Specifying Row and Column Level Security

This section provides an overview of role and user-based security and discusses how to:

- Define security roles.
- Grant dimension and metric access to security roles.
- Assign users to security roles.
- View summary for security roles.
- Grant dimension and metric access to a user.
- View summary for a secured user.
- Request security processing

Defining Dimension and Metric Security

Because EPM is delivered with no security restrictions, dimensions and metrics (also known as fact-columns) are also delivered unsecured. Before you can grant a user access to a dimension or metric, you must first indicate to the system that a particular dimension or metric requires securing. The pages used to define dimension and metric security are discussed in the security chapter of this PeopleBook.

See Chapter 6, "Setting Up EPM Security," Defining Dimension and Metric Security, page 128.

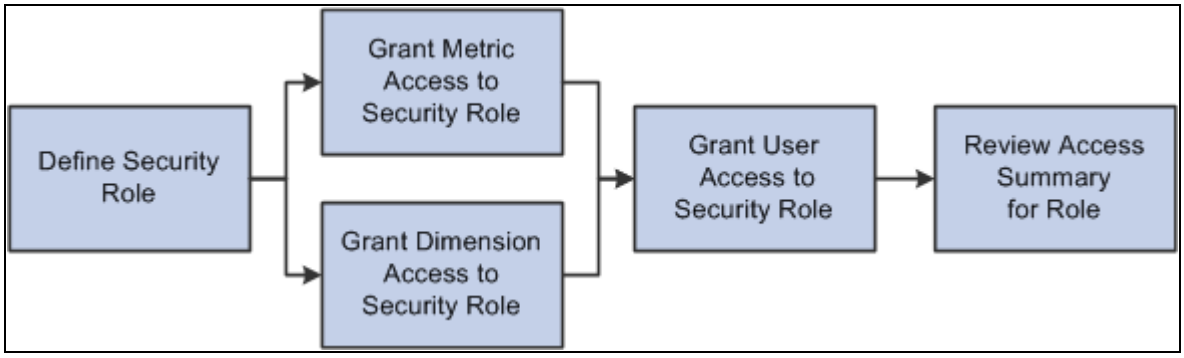
Note. Dimensions and metrics that are not secured are classified as public, or unsecured. All EPM users can view public objects.

Understanding Role and User Based Security for Dimensions and Metrics

After you designate dimensions and metrics that require securing, you must grant users access to those objects. You can grant security access to an individual user or to a specific security *role*.

EPM security enables you to create security roles. A security role is a set of data access privileges that are assigned to one or more users. A user who is assigned to a specific role inherits all access privileges that are associated with that role. A user can belong to multiple roles. In this case, the user would inherit the combined privileges that are defined for all roles.

To set up security roles, define the role, assign dimension and metric access privileges to the role, and then assign users to the role. The following diagram depicts this process.

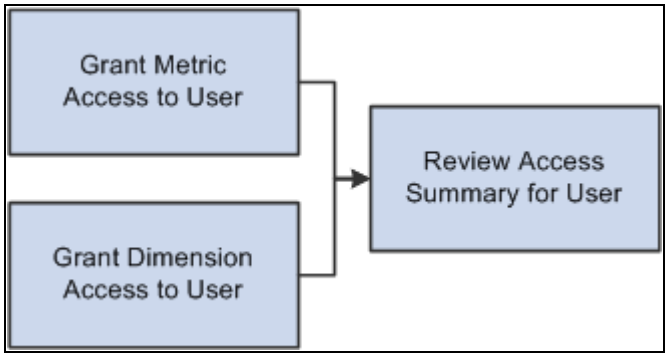


Process flow - security role setup

If you have established security roles for your PeopleTools security, you can import the roles into the EPM database using the Run Security Processing page.

See Chapter 6, "Setting Up EPM Security," Processing Your Security Parameters, page 145.

EPM security also enables you to define access privileges for individual users. To set up user access privileges, assign dimension and metric access privileges to a specific user.



Process flow - user security setup

Dimension Security and Individual, Constraint, and Tree Based Selections

EPM security provides three methods to specify row-level security for your dimension:

- Individual-based definition: Enables you to specify row-level security using the setID and dimension key fields.
- Constraint-based definition: Enables you to specify row-level security by associating a constraint with a dimension table.

  The constraint limits access to a dimension by acting as the WHERE clause in a SQL statement—for example, SELECT Account ID FROM Account Dimension WHERE Account ID = Northwest.

  Set ID and Constraint ID are used to specify constraint access.

- Tree-based definition: Enables you to specify row-level security using existing tree hierarchies that are defined for a dimension.

  You can use the tree to grant a user access to specific nodes, leaves, or details in the tree. Tree hierarchy use is limited to OWE tables only.

Processing Role and User Based Security Parameters

After the security rules have been set up, the EPM security application engine (PF_SECURITY) process must be run. This processes the access as defined in the Role Dimension access pages and flattens the data to the individual dimension members and populates the security join tables specified in dimension metadata.

Pages Used to Establish Role and User-Based Security

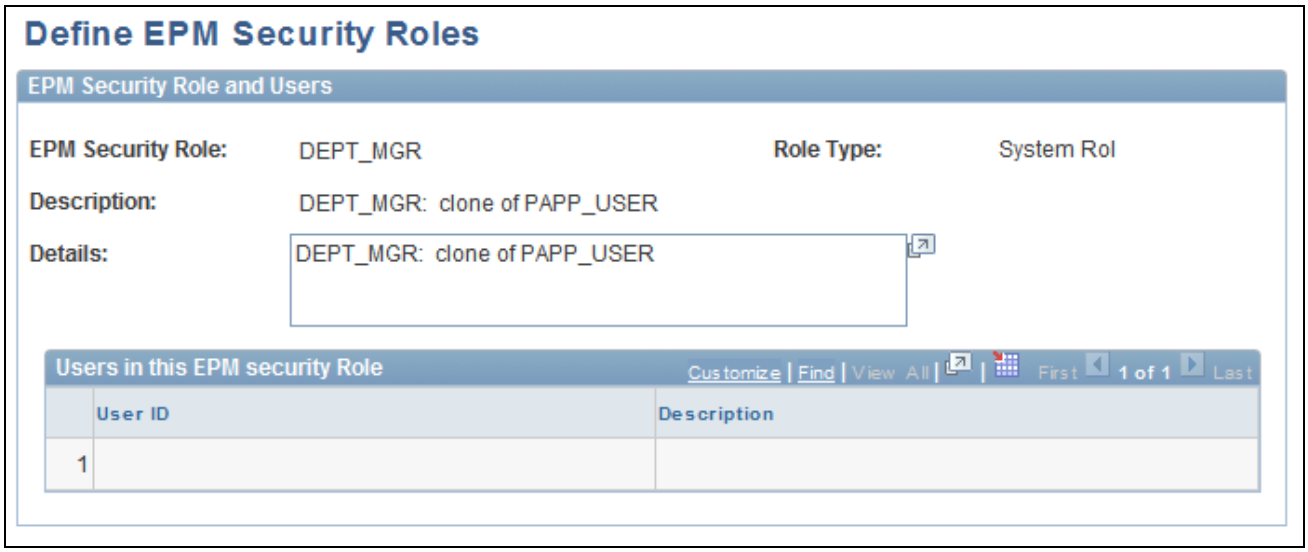
Page Name	*Definition Name*	*Navigation*	*Usage*
Define EPM Security Roles	PF_SY_ROLE_DEFN	EPM Foundation, EPM Security, Security By Role, Define EPM Security Roles	Define security roles.
Access To Metric	PF_SY_ROLE_METR	EPM Foundation, EPM Security, Security By Role, Role Metric Access, Access To Metric	Assign a metric to a security role.
Role Dimension Access	PF_SY_ROLE_ALL_MDW	EPM Foundation, EPM Security, Security By Role, Role Dimension Access, Role Dimension Access	Assign a dimension table to a security role and specify high-level access privileges. **Note.** The object name of this page changes depending on whether an OWE or MDW dimension is selected.
Individual Selection	PF_SY_ROLE_LIST	EPM Foundation, EPM Security, Security By Role, Role Dimension Access, Individual Selection	Specify row-level access to the dimension based on setID and dimension key fields.
Constraint-based Selection	PF_SY_ROLE_CONS	EPM Foundation, EPM Security, Security By Role, Role Dimension Access, Constraint-based Selection	Specify row-level access to the dimension based on constraint.

Page Name	*Definition Name*	*Navigation*	*Usage*
Select Security Column	PF_SY_COLUMN_PG	Automatically accessed when you select a constraint on the Constraint-based Selection page that contains two or more columns and the system cannot distinguish the column you want to include in your security parameters	Select one column to include in your constraint.
Tree-based Selection	PF_SY_ROLE_TREE	EPM Foundation, EPM Security, Security By Role, Role Dimension Access, Tree-based Selection	Specify row-level access to the dimension based on an existing tree hierarchy that is defined for the dimension. You can use the tree to grant a user access to specific nodes, leaves, or details in the tree. Used with OWE dimensions only.
User Role Access	PF_SY_USER_ROLES	EPM Foundation, EPM Security, Security By User, User Role Access	Assign a user to a security role.
Role Security Summary	PF_SY_ROL_SUMMARY	EPM Foundation, EPM Security, Security By Role, Role Security Summary	Review access privileges and other information for a security role.
Review Role Dimension Access	PF_SY_ROLE_DETAIL	Click the *Details* link on the Role Security Summary page.	Review additional details about a dimension that is associated with the selected security role
Access To Metric	PF_SY_ROLE_METR	EPM Foundation, EPM Security, Security By User, User Metric Access, Access To Metric	Assign a metric to a user.
User Dimension Access	PF_SY_ROLE_ALL	EPM Foundation, EPM Security, Security By User, User Dimension Access, User Dimension Access	Assign a dimension table to a user and specify high-level access privileges.
Individual Selection	PF_SY_ROLE_LIST	EPM Foundation, EPM Security, Security By User, User Dimension Access, Individual Selection	Specify row-level access to the dimension based on setID and dimension key fields.
Constraint-based Selection	PF_SY_ROLE_CONS	EPM Foundation, EPM Security, Security By User, User Dimension Access, Constraint-based Selection	Specify row-level access to the dimension based on constraint.

Page Name	Definition Name	Navigation	Usage
Select Security Column	PF_SY_COLUMN_PG	Automatically accessed when you select a constraint on the Constraint-based Selection page that contains two or more columns and the system cannot distinguish the column you want to include in your security parameters.	Select one column to include in your constraint.
Tree-based Selection	PF_SY_ROLE_TREE	EPM Foundation, EPM Security, Security By User, User Dimension Access, Tree-based Selection	Specify row-level access to the dimension based on an existing tree hierarchy that is defined for the dimension. You can use the tree to grant a user access to specific nodes, leaves, or details in the tree. Used with OWE dimensions only.
User Security Summary	PF_SY_SUMMARY	EPM Foundation, EPM Security, Security By User, User Security Summary	Review access privileges and other information for a user.
Review User Dimension Access	PF_SY_USER_DETAIL	Click the *Details* link on the User Security Summary page.	Review additional details about a dimension that is associated with the selected user.
Request Security Processing	RUN_PF_SECURITY	EPM Foundation, EPM Security, Advanced, Request Security Processing	Apply security parameters for security roles and users by running the request security processing (PF_SECURITY) process.

Defining Security Roles

Access the Define EPM Security Roles page (EPM Foundation, EPM Security, Security By Role, Define EPM Security Roles).



Define EPM Security Roles page

EPM Security Role Displays the name of the security role that you are defining.

Role Type Displays the type of role that is being defined.

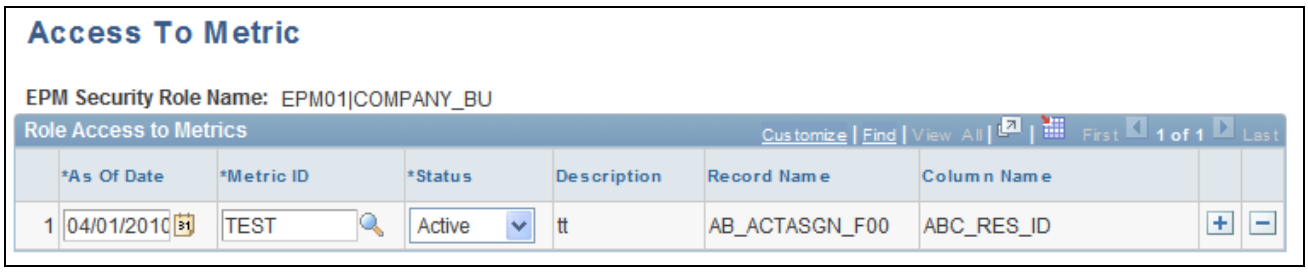
Possible role types include *EPM Role, System Role, User Role, WFA Generated Role.*

Users in This EPM Security Role

User ID Displays the users who are associated with this role.

Granting Metric Access to a Security Role

Access the Access To Metric page (EPM Foundation, EPM Security, Security By Role, Role Metric Access, Access To Metric).



Access To Metric page

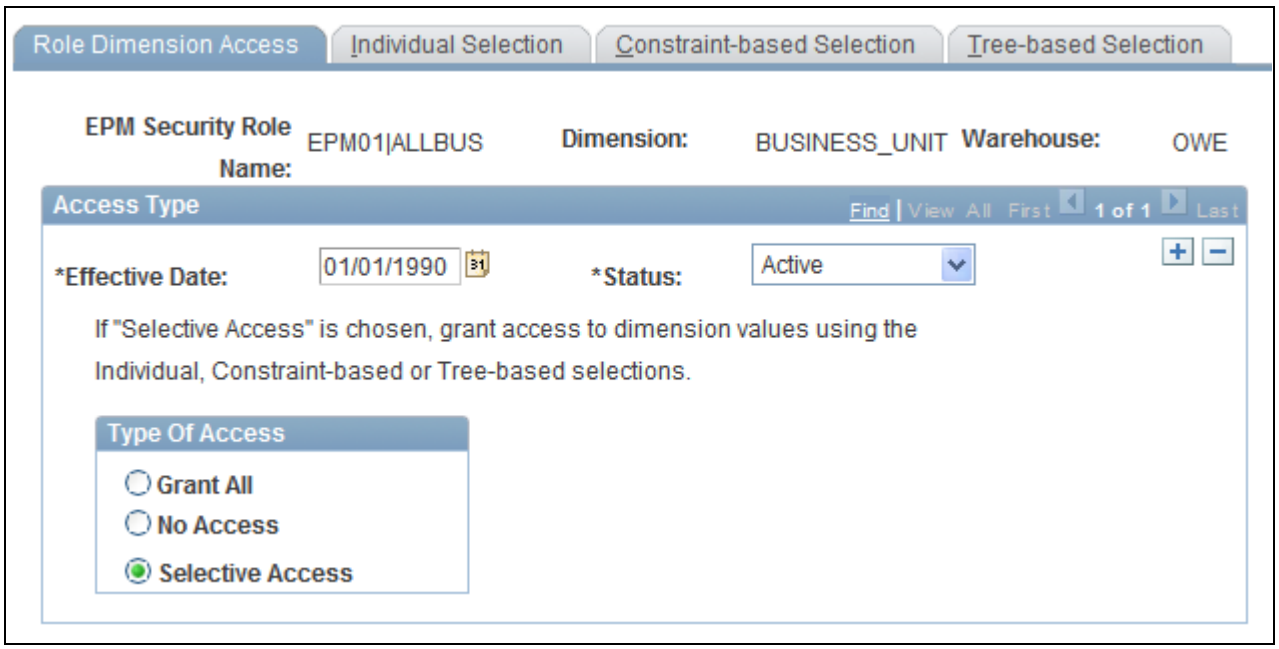
Metric ID Displays the metric that you are associating with a particular security role.

Record Name Displays the record that is associated with the selected metric.

Column Name Displays the column that is associated with the selected metric.

Granting Dimension Access to a Security Role

Access the Role Dimension Access page (EPM Foundation, EPM Security, Security By Role, Role Dimension Access, Role Dimension Access).



Role Dimension Access page

Dimension Displays the dimension that you are associating with a particular security role.

Warehouse Displays the warehouse layer that is associated with the selected dimension.

Type of Access

Grant All Select this option to grant the role access to the entire dimension.

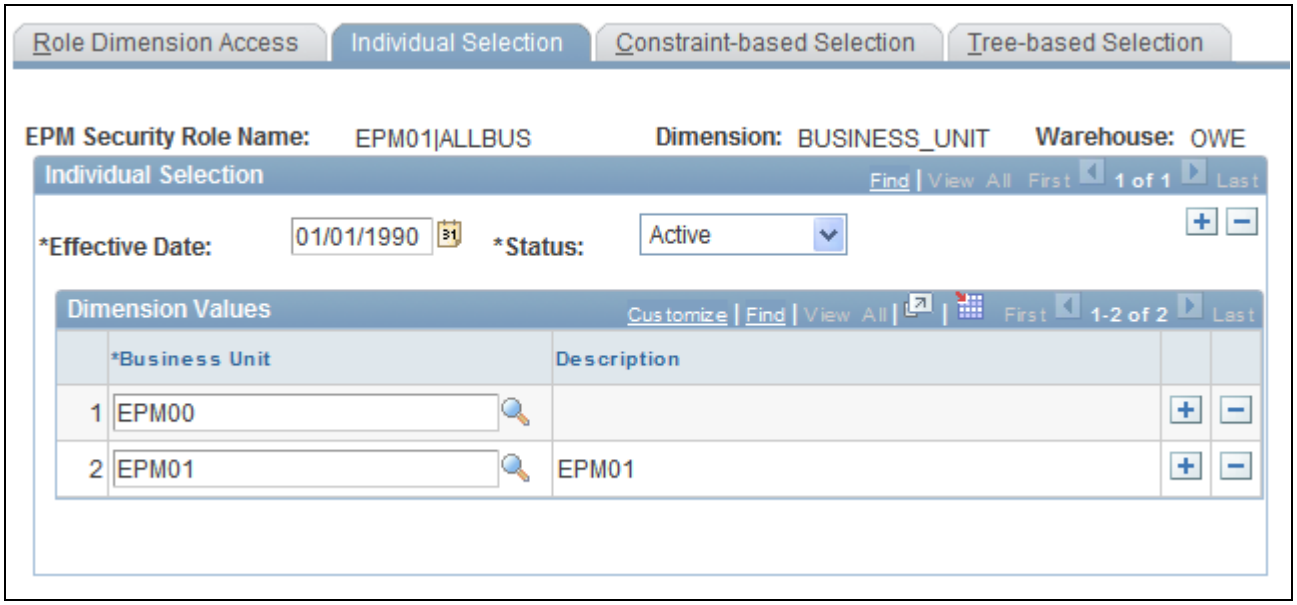
No Access Select this option to bar the role from accessing the entire dimension.

Selective Access Select this option to grant the role access to specific rows in the dimension.

You can specify rows individually based on setIDs, using a constraint or using a hierarchy tree that is defined for the dimension (tree hierarchies are available only for OWE dimensions) .

Granting Individual Row-Level Access to a Role

Access the Individual Selection page (EPM Foundation, EPM Security, Security By Role, Role Dimension Access, Individual Selection).



Individual Selection page

Dimension Displays the dimension that you are associating with a particular security role.

Warehouse Displays the warehouse layer that is associated with the selected dimension.

Dimension Values

SetID or Business Unit Enter the setID or business unit that is associated with the dimension rows that you want to secure.

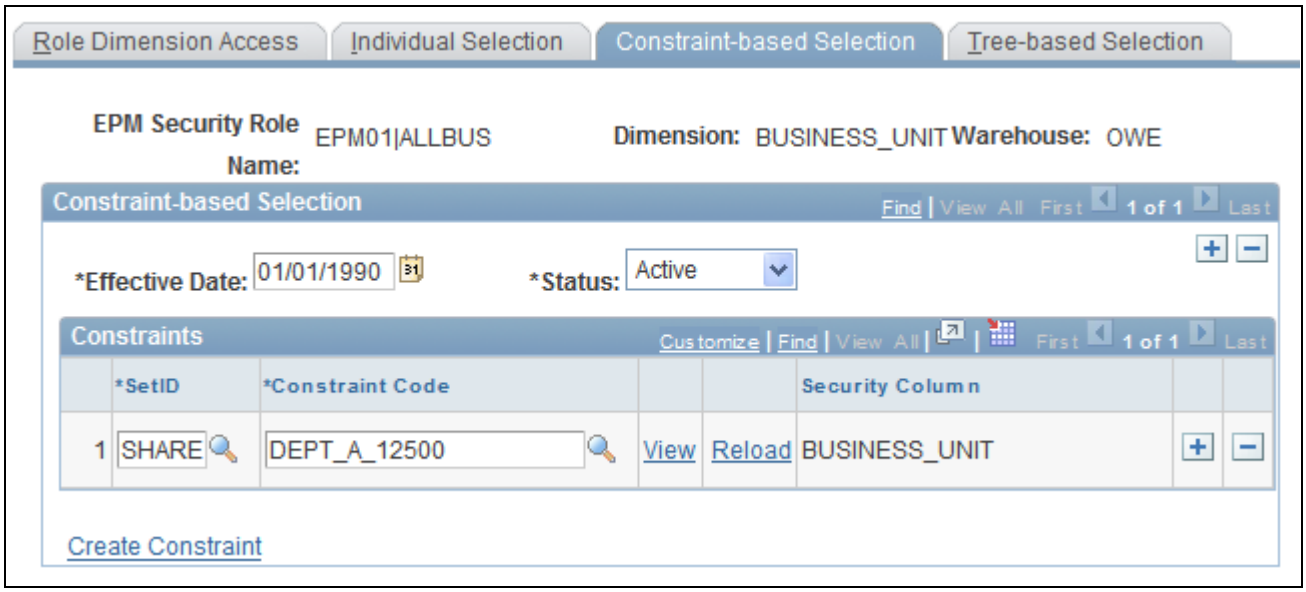
This field can display either SetID or Business Unit, depending on the dimension you select. In some instances, there is no value displayed for the field .

Dimension Key Enter the dimension key that is associated with the dimension.

Because this is a dimension key field, the name of this field changes depending on the selected dimension. For example, if the Product (PRODUCT) table were selected, *Product ID*would be displayed because it is the dimension key for that table.

Granting Constraint-Based Row-Level Access to a Role

Access the Constraint-based Selection page (EPM Foundation, EPM Security, Security By Role, Role Dimension Access, Constraint-based Selection).



Constraint-based Selection page

Dimension Displays the dimension that you are associating with a particular security role.

Warehouse Displays the warehouse layer that is associated with the selected dimension.

Constraint-based Selection

SetID Enter the setID that is associated with the dimension rows that you want to secure.

Constraint Code Enter the constraint that you want to associate with the selected dimension rows.

You must have a constraint defined before you can access it here. If you do not have a constraint defined, you can use the Create Constraint link to create a new constraint.

Note. The Select Security Column page displays if the constraint you select contains two or more columns and the system cannot distinguish the column you want to include in your security parameters.

Reload Click to refresh the constraint definition if you have changed it after it was included in a security role.

Security Column Displays the field from the constraint that is used as the column to restrict access.

Create Constraint Click to access the Constraints page and define a constraint.

If you have not created a constraint for the selected dimension rows, you can do so in the Constraints page.

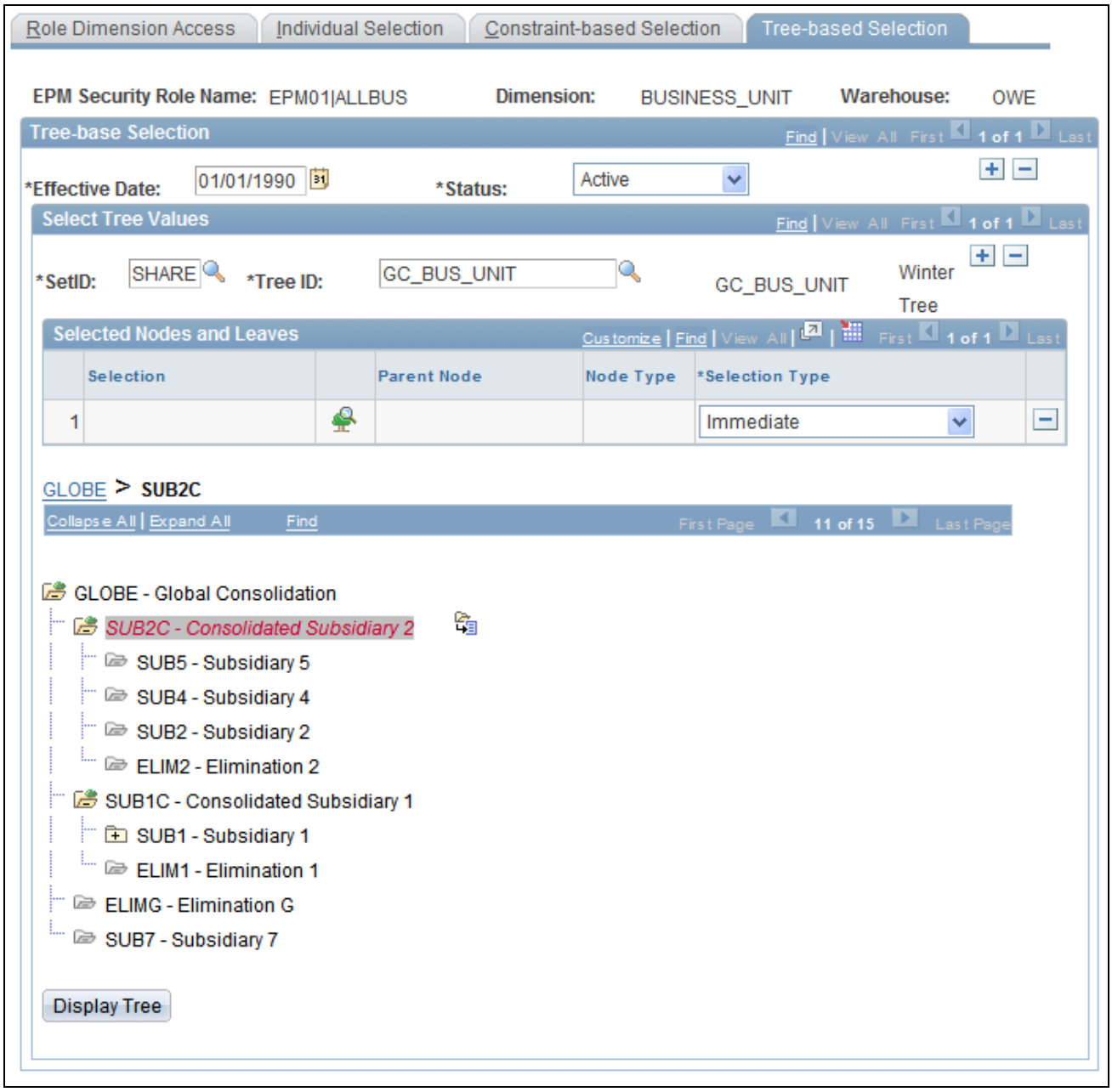
Specifying Constraint Columns for Constraint-Based Row-Level Access

Access the Select Security Column page (Automatically accessed when you select a constraint on the Constraint-based Selection page that contains two or more columns and the system cannot distinguish the column you want to include in your security parameters).

When you select a constraint (on the Constraint-based Selection page) that contains two or more columns and the system cannot distinguish the column you want to include in your security parameters, the Select Security Column page is accessed automatically. The page displays the columns available to use in the constraint you selected. You must choose just one of the columns for the constraint. Select the column you want to include by clicking the column name in the Key ID field.

Granting Tree-Based Row-Level Access to a Role (OWE Dimension Only)

Access the Tree-based Selection page (EPM Foundation, EPM Security, Security By Role, Role Dimension Access, Tree-based Selection).



Tree-based Selection page

Dimension Displays the dimension that you are associating with a particular security role.

Warehouse Displays the warehouse layer that is associated with the selected dimension.

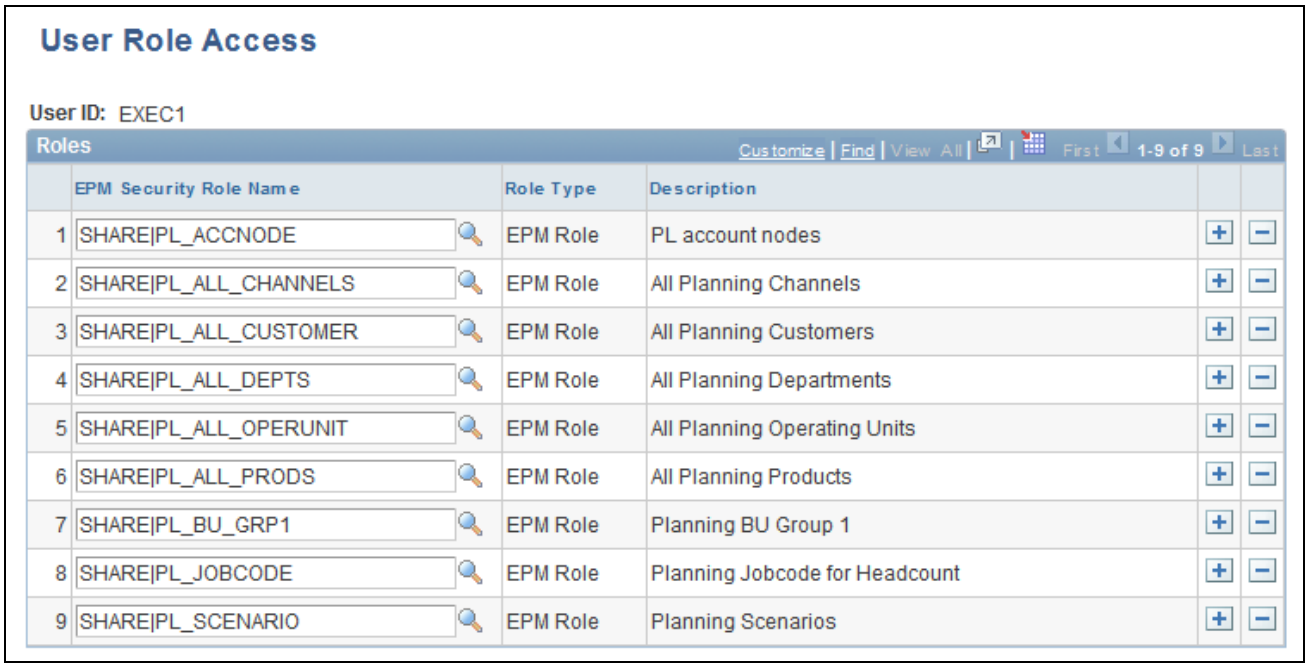
Select Tree Values

SetID Enter the setID that is associated with the dimension rows that you want to secure.

Tree ID	Enter the hierarchy tree that you want to use to specify the dimension rows.
Selection	Displays the selected tree node value.
Find Selected Value	Click the Find Selected Value button to display the selected node at the top of the hierarchy tree and make it easier for you to locate the node with which you are working.
Parent Node	Displays the parent node of the selected node. This field is blank if the selected node is a root node.
Node Type	Displays the node type of the selected node. Values can be *Node* or *Detail.*
Selection Type	Specify the level of detail to include with the selected node. Different values are available for your selection, depending on whether you have selected a node or a leaf from the hierarchy tree. If a node is selected, you can specify *This Node Only, Immediate Children, Node and Immediate Children, All Descendants,* or *Node + All Descendants.* If a leaf is selected, you can specify *Immediate Child Leaves* or *All Descendant Leaves.* If the leaf has a range of values, you must select *Immediate Child Leaves.* Trees with duplicate leaves are not supported.
Add to Node Selection List	Click the Add to Node Selection List button to add the selected node to the selection list. You must add a node to the selection list before the fields in the Selected Nodes and Leaves group box displays node values.
Display Tree	Click to display the hierarchy tree.

Assigning Users to a Security Role

Access the User Role Access page (EPM Foundation, EPM Security, Security By User, User Role Access).



User Role Access

User ID: EXEC1

Roles — Customize | Find | View All | First 1-9 of 9 Last

	EPM Security Role Name	Role Type	Description
1	SHARE|PL_ACCNODE	EPM Role	PL account nodes
2	SHARE|PL_ALL_CHANNELS	EPM Role	All Planning Channels
3	SHARE|PL_ALL_CUSTOMER	EPM Role	All Planning Customers
4	SHARE|PL_ALL_DEPTS	EPM Role	All Planning Departments
5	SHARE|PL_ALL_OPERUNIT	EPM Role	All Planning Operating Units
6	SHARE|PL_ALL_PRODS	EPM Role	All Planning Products
7	SHARE|PL_BU_GRP1	EPM Role	Planning BU Group 1
8	SHARE|PL_JOBCODE	EPM Role	Planning Jobcode for Headcount
9	SHARE|PL_SCENARIO	EPM Role	Planning Scenarios

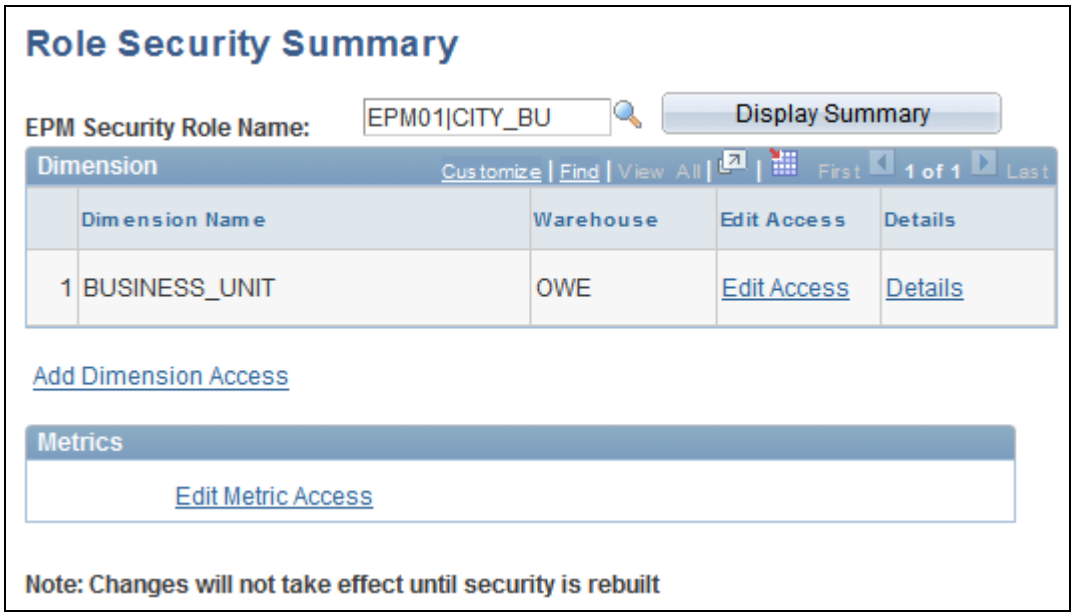
User Role Access page

User ID Displays the user for whom you are granting role access.

EPM Security Role Name Enter the security role that you want to associate with the selected user.

Viewing Security Summary for a Security Role

Access the Role Security Summary page (EPM Foundation, EPM Security, Security By Role, Role Security Summary).



Role Security Summary page

EPM Security Role Name Enter the security role for which you want to see a summary of access privileges.

Display Summary Click to display the security role details and refresh the view.

Dimension

Dimension Name Displays the dimensions that are associated with the selected security role.

Warehouse Displays the warehouse layer that is associated with the selected dimension.

Edit Access Click to access the Role Dimension Access page and edit the security role's access to the dimension.

Details Click to access the Review Role Dimension Access page and examine additional details about the secured dimension, such as the setID or dimensionID.

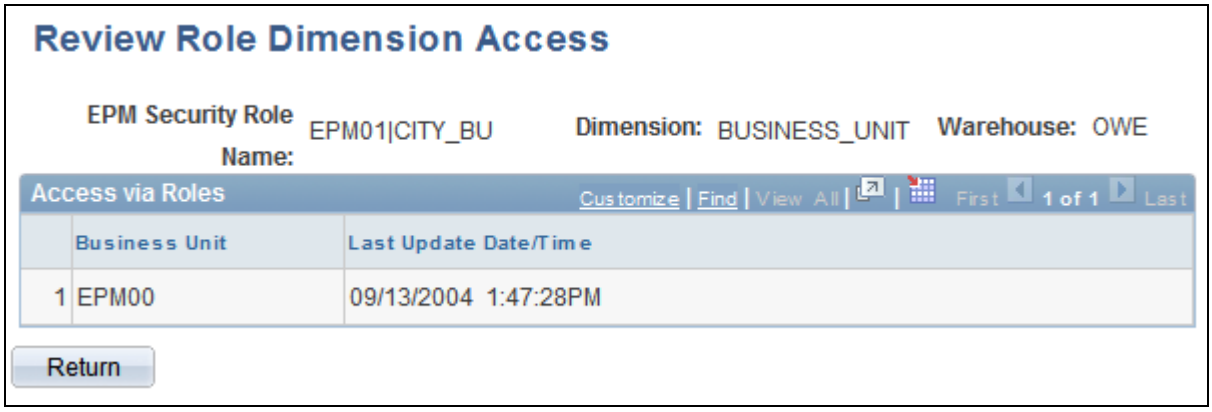
Add Dimension Access Click to access the Role Dimension Access page and grant the selected security role access to another dimension.

Metrics

Click the Edit Metric Access link to access the Role Metric Access page and edit the security role's access to a metric.

Review Role Dimension Access Summary

Access the Review Role Dimension Access page (Click the Details link on the Role Security Summary page.).



Review Role Dimension Access page

Use this page to review additional details about your dimension that is associated with a particular security role.

Granting Metric Access to a User

Access the Access to Metric page (EPM Foundation, EPM Security, Security By User, User Metric Access, Access To Metric).

The fields on this page are identical to the fields on the Access to Metric page for security roles. The only difference is that the fields on this page represent individual user access privileges and not a security role.

Granting Dimension Access to a User

Access the User Dimension Access page (EPM Foundation, EPM Security, Security By User, User Dimension Access, User Dimension Access).

The fields on this page are identical to the fields on the Role Dimension Access page. The only difference is that the fields on this page represent individual user access privileges and not a security role.

Granting Individual Row-Level Access to a User

Access the Individual Selection page (EPM Foundation, EPM Security, Security By User, User Dimension Access, Individual Selection).

The fields on this page are identical to the fields on the Individual Selection page for security roles. The only difference is that the fields on this page represent individual user access privileges and not a security role.

Granting Constraint-Based Row-Level Access to a User

Access the Constraint-based Selection page (EPM Foundation, EPM Security, Security By User, User Dimension Access, Constraint-based Selection).

The fields on this page are identical to the fields on the Constraint-based Selection page for security roles. The only difference is that the fields on this page represent individual user access privileges and not a security role.

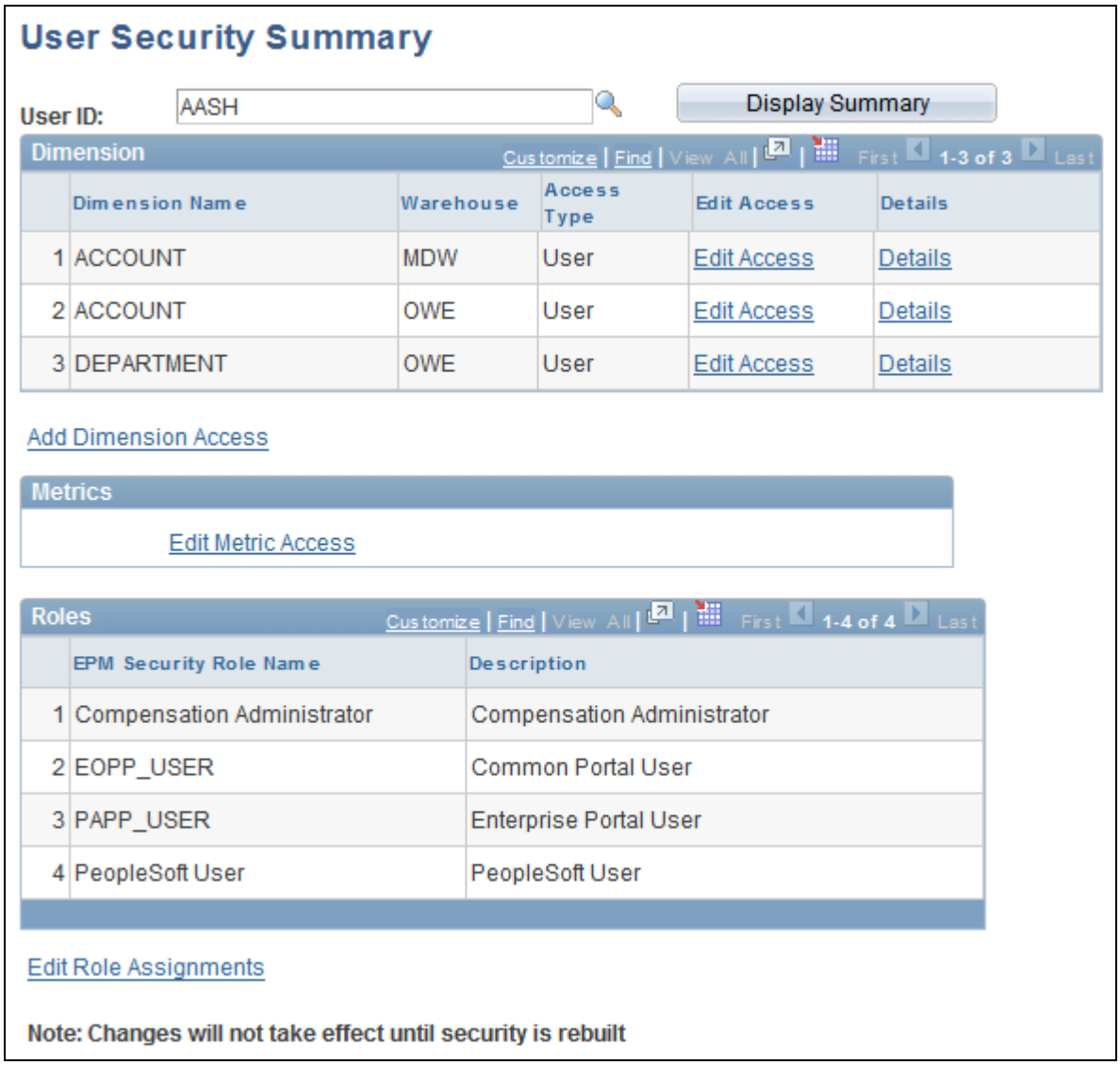
Granting Tree-Based Row-Level Access to a User (OWE Dimension Only)

Access the Tree-based Selection page (EPM Foundation, EPM Security, Security By User, User Dimension Access, Tree-based Selection).

The fields on this page are identical to the fields on the Tree-based Selection page for security roles. The only difference is that the fields on this page represent individual user access privileges and not a security role.

Viewing Security Summary for a User

Access the User Security Summary page (EPM Foundation, EPM Security, Security By User, User Security Summary).



User Security Summary page

User ID Enter the user for which you want to see a summary of access privileges.

Display Summary Click to display security details for the user and refresh the view.

Dimension

Dimension Name Displays the dimensions that are associated with the selected user.

Warehouse Displays the warehouse layer that is associated with the selected dimension.

Edit Access Click to access the User Dimension Access page and edit the user's access privileges to the dimension.

Details	Click to access the Review User Dimension Access page and examine additional details about the secured dimension, such as the setID or dimensionID.
Add Dimension Access	Click to access the User Dimension Access page and grant the selected user access to another dimension.

Metrics

Click the Edit Metric Access link to access the User Metric Access page and edit the user's access to a metric.

Roles

EPM Security Role Name	Displays the security roles that are associated with selected user.
Edit Role Assignments	Click to access the User Role Access page and edit the user's privileges that are associated with the role.

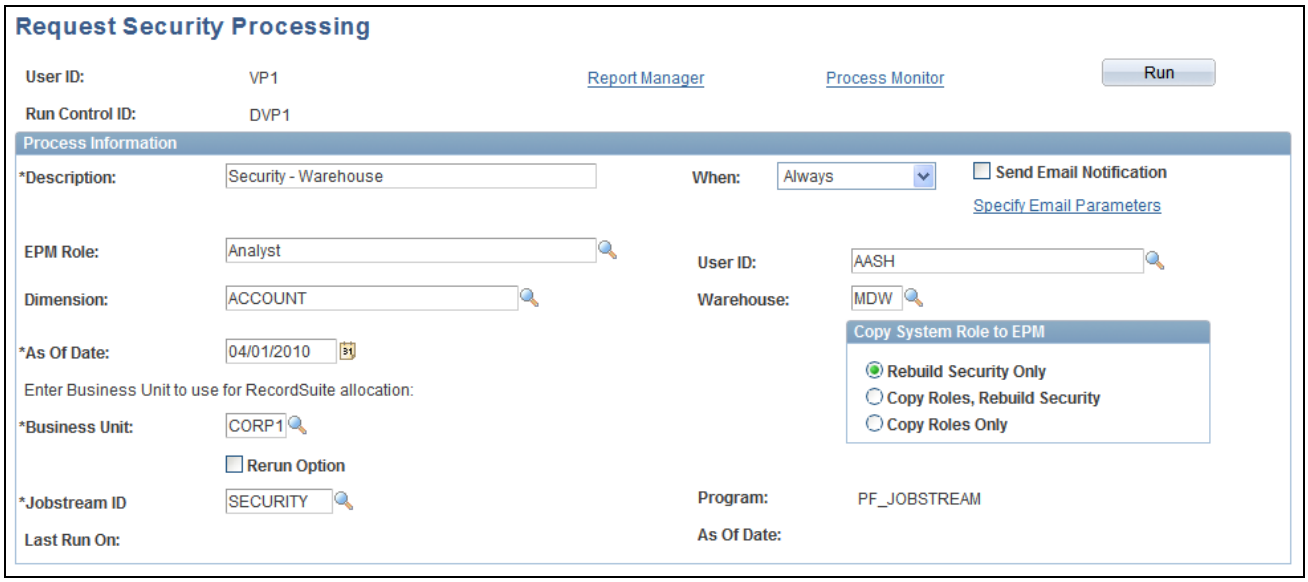
Review User Dimension or User Metric Access Summary

Access the Review User Dimension Access page (Click the Details link on the User Security Summary page).

The fields on this page are identical to the fields on the Review Role Dimension Access page. The only difference is that the fields on this page represent individual user access privileges and not a security role.

Processing Your Security Parameters

Access the Request Security Processing page (EPM Foundation, EPM Security, Advanced, Request Security Processing).



Request Security Processing page

EPM Role Enter the EPM security role that you want to process.

If you leave this field blank, all security roles are processed.

Note. You cannot process a security role and a user at the same time.

User ID Enter the user you want to process.

If you leave this field blank, all users are processed.

Note. You cannot process a security role and a user at the same time.

Dimension Enter the dimension that you want to process.

If you leave this field blank, all dimensions are processed.

Warehouse Enter the warehouse structure that is associated with the dimension you select for processing.

Business Unit Enter the business unit that you want to process.

Business unit is used to determine which record suite is used for the security job.

Jobstream ID Enter the Jobstream ID for the warehouse security.

Note. This jobstream is not secured, all users can access and run it. However, only an adiministrator should run this jobstream.

Rerun Option Select this check box to rerun the security parameters process.

Copy System Role to EPM

Rebuild Security Only	Select this option if you want only to rebuild the security join tables.
Copy Roles, Rebuild Security	Select this option if you want to rebuild the security join tables and import PeopleTools security roles into the EPM database.
Copy Roles Only	Select this option if you want only to import PeopleTools security roles into the EPM database.

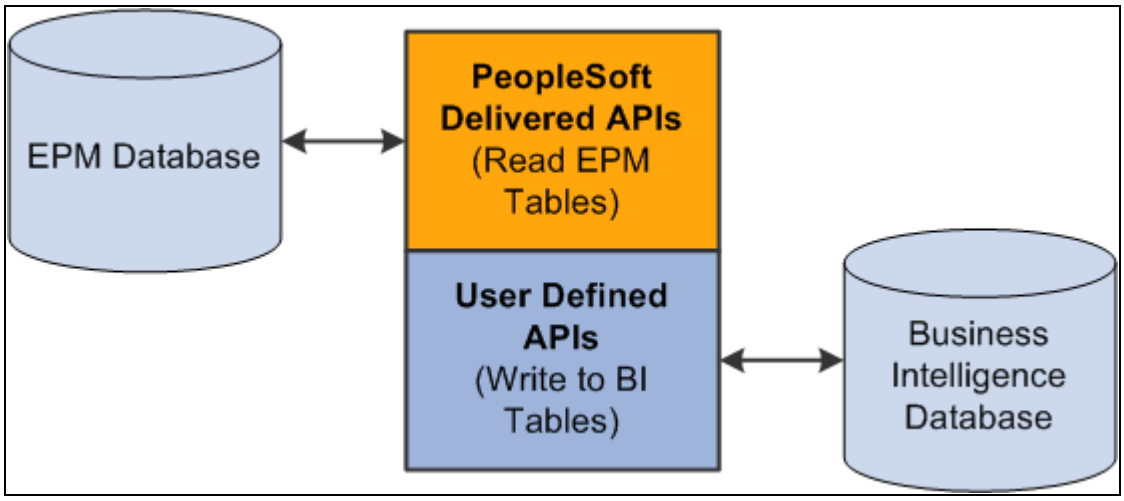
Establishing Security Bridges for Your Business Intelligence Tool

This section provides and overview of security bridges and discusses PeopleSoft delivered Application Programming Interfaces (APIs).

Understanding Security Bridges

Security bridges provide a means to transfer EPM security profiles to your Business Intelligence (BI) database. This enables you to define your security parameters once in EPM without having to redefine them in your BI reporting tool.

Security bridges use Application Programming Interfaces (APIs) to transfer security profiles. APIs use programming commands to interface and communicate with your BI database and transfer EPM security profiles. A set of APIs is required to read data from EPM tables and another set is required to write data to your BI tables.



Security bridge and APIs

PeopleSoft prepackages APIs that are designed to read data from your EPM tables. However, you must create the code that writes data to your BI tables. PeopleSoft APIs are coded in Java and can function with any BI reporting tool.

PeopleSoft delivers the following APIs.

Role Collection

This class is a collection class of individual Role classes. The collection will inherit the normal Java methods to traverse the collection, for example, next(), first() and so forth. Also methods such as add(), remove() can be used and implemented as an ArrayList.

Public Methods

Name RoleCollection(JDBCConnection dbCon)

Returns Nothing

Parameters JDBCConnection dbCon. A valid JDBC Connection.

Description This is the constructor used to instantiate the Role collection.

Name LoadAllRoles()

Returns Nothing

Parameters None

Description This method clears out the current elements in the collection and load it with all the currently defined Roles in EPM.

Role

This class defines a single Role defined in the EPM row-level security framework.

Public Methods

Name Role(JDBCConnection dbCon, String Name);

Returns Nothing

Parameters JDBCConnection dbCon. A valid JDBC Connection.

Description The constructor for the Role class. Takes JDBC Connection and RoleName as parameters. Given a rolename, the constructor will fill up the other properties such as Type and Description.

Note. There is no public constructor of the format Role(), you cannot instantiate this class without a name.

Name	getUsers();
Returns	List of UserNames
Parameters	None
Description	The method will return a List of User Names that belong to this Role. A standard Java List object can be used to implement this.

Name	getAccessibleDimensions();
Returns	List of Dimension Names
Parameters	None
Description	The method will return a List Dimension names that this Role can access.

Name	hasAccess (String DimName);
Returns	integer 0 or 1
Parameters	Dimension Name
Description	The method returns a Boolean specifying whether the Dimension referenced by DimensionName parameter is available to the Role or not.

Name	hasAccess(String DimName, int DimValue);
Returns	integer 0 or 1
Parameters	Dimension Name, A surrogate key value
Description	The method returns 0 or 1 specifying whether this Role has access to a particular value in a dimension.

Name	hasAllAccess(String DimName);
Returns	integer 0 or 1
Parameters	Dimension Name
Description	The method returns a Boolean specifying whether this Role has ALL access to the Dimension referenced by DimensionName parameter.

Name	getDimensionValues(String DimName);

Returns	List of Values
Parameters	Dimension Name
Description	The method returns a List of values in the Dimension referenced by DimName that are available to the Role. The method will first check to ensure that the Dimension Name is a valid Secured dimension in the system, and it is accessible to the Role. The method will retun a List of values from the appropriate security join table. If the Role has ALL access to this dimension, the List will contain only one value with the pre-determined surrogate key value for ALL. Will return an empty List if the dimension is not available to the Role.

Name	getName();
Returns	String Name
Parameters	None
Description	Get method for Name

Name	getType();
Returns	String Name
Parameters	None
Description	Get method for Type

Name	getDescription();
Returns	Stirng Descr
Parameters	None
Description	Get method for Description

Name	getLongDescription();
Returns	String Description
Parameters	None
Description	Getter method for Long Description

User Collection

This class contains the same information as the Role Collection class. The only difference is that the values for this class represent user collection and not role collection.

User

This class contains information about a single User defined in EPM Security.

Public Methods

Name	User(JDBCConnection dbCon, String Name);
Returns	Nothing
Parameters	JDBC Connection, String UserName
Description	The constructor for the User class. Takes Name as the one and only parameter.

Name	getRoles();
Returns	List of RoleNames
Parameters	None
Description	The method will instantiate a List of Role Names that contain this User.

Name	getAccessibleDimensions();
Returns	List of Dimension Names
Parameters	None
Description	The method will return a List of Dimension names that this User can access. This will be a combined list of all dimensions that the user can access via all his roles.

Name	hasAccess (String DimName); getAccessibleMetrics()
Returns	integer 0 or 1
Parameters	Dimension Name

Description	The method returns a Boolean specifying whether the Dimension referenced by DimensionName parameter is available to the User or not. This method queries all the Roles for this user to determine if the user has access to the dimension.
Name	hasAccess(String DimName, int DimValue);
Returns	integer 0 or 1
Parameters	Dimension Name, A surrogate key value
Description	The method returns 0 or 1 specifying whether this User has access to a particular value in a dimension. This method queries all the Roles for this user to determine if the user has access to the dimension and value.
Name	hasAllAccess(String DimName);
Returns	integer 0 or 1
Parameters	Dimension Name
Description	The method returns a Boolean specifying whether this User has ALL access to the Dimension referenced by DimensionName parameter. If any of the user's Roles has ALL access, the user is deemed to have ALL access.
Name	getDimensionValues(String DimName);
Returns	List of Values
Parameters	Dimension Name
Description	The method returns a List of values in the Dimension referenced by DimName that are available to the User. The method first checks to ensure that the Dimension Name is a valid Secured dimension in the system, and it is accessible to the User. This is a combined list of values from all the user's Roles. The method returns a List of values from the appropriate SJT. If the User has "all" access to this dimension, the List will contain only one value with the pre-determined surrogate key value for "all". Will return an empty List if the dimension is not available to the User.
Name	getName();
Returns	String Name
Parameters	None
Description	Get method for Name

Dimension Collection

This class contains information about the list of dimensions defined as secured dimensions in EPM Security. Each element of the collection is a Dimension class that represents one secured Dimension in the EPM system. The collection will inherit the normal Java methods to traverse the collection, for example, next(), first() and so forth. Also methods such as add(), remove() can be used.

Public Methods

Name	DimensionCollection(JDBCConenction dbCon)
Returns	Nothing
Parameters	JDBC Connection
Description	This is the constructor used to instantiate the Dimension collection. This class is derived from one of the Java Collection classes, possibly ArrayList, or LinkedList. Most of the methods needed are inherited from the Collection class, Next(), Previous(), HasNext(), size() and so forth. The constructor will instantiate an empty collection.

Name	LoadAllDimensions()
Returns	Nothing
Parameters	None
Description	This method will clear out the current elements in the collection and load it with all the currently secured dimensions in EPM.

Dimension

This class contains information about a single Dimension defined in the EPM database.

Public Methods

Name	Dimension (JDBC Connection, String Name)
Returns	String Name
Parameters	JDBC Connection

Description	This is the constructor for this class. Takes a Dimension Name as parameter. The constructor will validate that the dimension specified by Name is a valid secured dimension in EPM. If it is an invalid name, or the dimension is not secured, the constructor will fail.

Name	getRoles()
Returns	List of Role Names
Parameters	Nothing
Description	This method will return a list of all Role names that have access to this dimension.

Name	getUsers()
Returns	List of User Names
Parameters	Nothing
Description	This method will return a list of all User names that have access to this dimension.

Name	getDimName()
Returns	Stirng Name
Parameters	None
Description	Get method for Name.

Name	getDimTableName()
Returns	String TableName
Parameters	None
Description	Get method for Table Name.

Name	getDimKeyName()
Returns	String KeyName
Parameters	None

Description	Get method for Key Name.

Name	isSecured()
Returns	Boolean
Parameters	None
Description	Get method for isSecured.

MetricCollection

This class contains information about metric collection.

Public Methods

Name	MetricCollection(JDBCConnection dbCon)
Returns	Nothing
Parameters	A valid JDBCConnection
Description	This is the constructor used to instantiate the Metric collection. The constructor will instantiate an empty collection.

Name	LoadAllMetrics()
Returns	Nothing
Parameters	None
Description	This method will clear out the current elements in the collection and load it with all the currently secured metrics in EPM.

Metric

This class contains information about the metrics (columns) setup in EPM Security. This is for the Column-level security.

Public Methods

Name	Metric(JDBCConnection dbCon , String Name)

Returns	Nothing
Parameters	JDBCConnection dbCon , String Name
Description	This is the constructor for this class. Takes a Dimension Name as parameter. There is no other public constructor. You need at least a Dimension Name to instantiate this class. The constructor will validate that the dimension specified by Name is a valid secured dimension in EPM. If it is an invalid name, or the dimension is not secured, the constructor will fail.

Name	getColName()
Returns	String Column Name
Parameters	None.
Description	This method will return the column name that the metric defines.

Name	getRecName()
Returns	String Record Name
Parameters	None
Description	This method will return the Record name that the metric defines.

JDBCConnection

This class is used to connect to a database using a JDBC driver.

Public Methods

Name	makeConnection()
Returns	Nothing
Parameters	None
Description	This method will connect to a JDBC database. The parameters needed to connect will be read from jdbcconnection.property file.

Name	makeConnection(String theDriverName, String theDbURL)
Returns	Nothing

Parameters	Driver Name, DB URL string
Description	This method will connect to a JDBC database. The parameters needed to connect are passed into the method.

Name	makeConnection(String theDriverName, String theDbURL, String theUserName, String thePassword)
Returns	Nothing
Parameters	JDBC Driver Name to be used to attempt the connection, URL for database to connect, User Name, Password
Description	This method will connect to a JDBC database. The parameters needed to connect are passed into the method.

Name	closeConnection()
Returns	Nothing
Parameters	None
Description	This method will close the connection.

Name	getDbResultSet(String pTheQuery)
Returns	ResultSet
Parameters	Query sent to the database.
Description	This method will execute the query and return a ResultSet object.

Part 3

Bringing Source Data Into EPM Using Extract, Transform, and Load (ETL)

Chapter 7

Preparing to Load Source Data Into EPM

This chapter provides an overview of the extract, transform, and load (ETL) process within EPM and discusses:

- IBM WebSphere DataStage
- ETL load strategies in EPM
- Data validation and error handling in the ETL process
- OWE Jobs
- MDW Jobs

See Also

Chapter 2, "Understanding PeopleSoft Enterprise Performance Management," page 5

Appendix A, "ETL Installation and Implementation Prerequisites and Considerations," page 639

Understanding ETL in EPM

This section discusses:

- ETL and the EPM Architecture.
- Data Flow through EPM.
- IBM WebSphere DataStage.

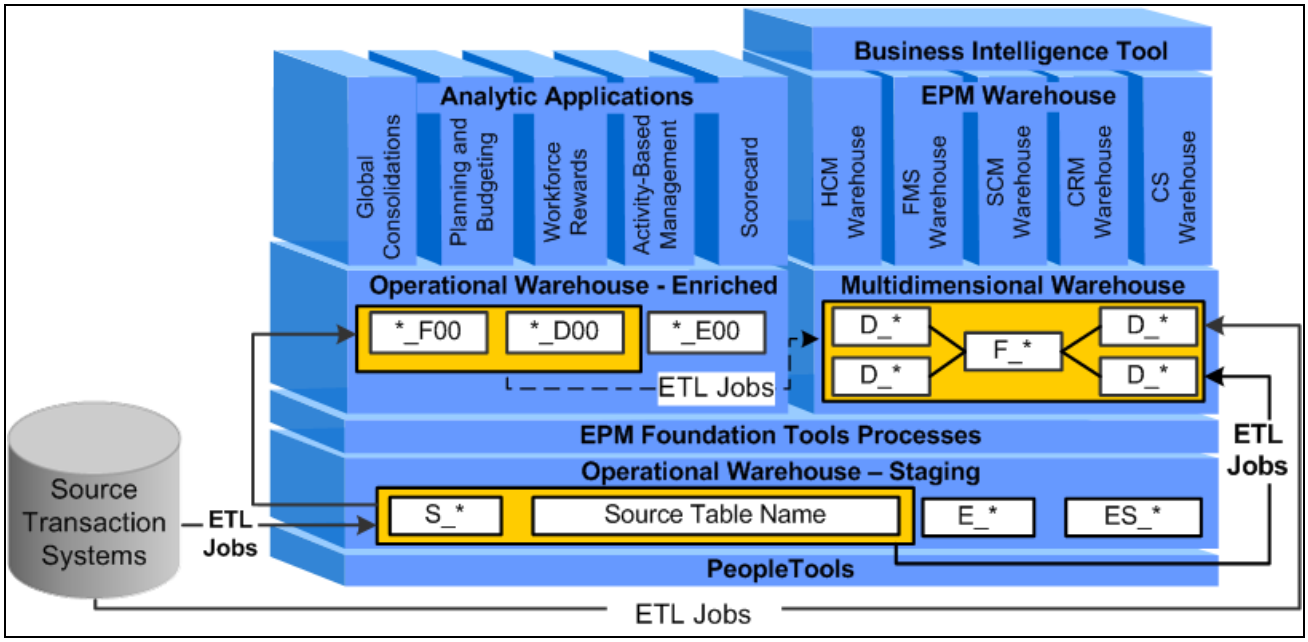
ETL and the EPM Architecture

The PeopleSoft delivered ETL process enables you to extract data from disparate source transaction systems, integrate the data in a single EPM database, transform and enrich the data, and load it into specific EPM data models that are optimized for analysis and reporting. This process is facilitated by the best-in-class data integration platform *IBM WebSphere DataStage* and PeopleSoft delivered ETL *jobs.*

The ETL process migrates data across all layers of EPM warehouse structures and consists of two load types:

- *Stage I Load*: Consists of all ETL jobs that extract data from your source transaction system and load it into Operational Warehouse - Staging (OWS) tables. Also included in this type of load (but less common) are ETL jobs that extract data from your source transaction system and load it directly into Multidimensional Warehouse (MDW) tables.
- *Stage II Load*: Consists of all ETL jobs that extract data from the OWS tables and load it into the Operational Warehouse - Enriched (OWE) or the Multidimensional Warehouse (MDW) tables. Also included in this type of load (but less common) are ETL jobs that extract data from the OWE and load it into the MDW.

The following diagram depicts the flow of data through each layer of the EPM architecture using ETL.



ETL in EPM

After your data is extracted from the OWS it is loaded into specialized data models (target warehouse tables designed to aggregate or enrich your data), which are used by the Analytical Applications and EPM Warehouses for reporting and analysis.

Understanding the Flow of Data Through EPM

Each EPM data warehouse requires a unique set of ETL jobs to populate corresponding target tables with data. Data warehouse target tables may have missing or inaccurate data in them if you do not run all applicable jobs in the proper sequence.

The following sections provide an overview of the ETL jobs required to populate each data warehouse layer with data.

Moving Data Into the OWS

You use ETL jobs to move data into the OWS from your PeopleSoft source system. The following is an overview of the steps required to bring data into the OWS:

1. Run initial setup (OWS) jobs.

2. Run source business unit extract jobs.
3. Run shared lookup jobs.
4. Run CSW OWS jobs (for CSW Warehouse implementation only).

   Run CRM OWS jobs (for CRM Warehouse implementation only).

   Run FMS OWS jobs (for FMS Warehouse implementation only).

   Run HCM OWS jobs (for HCM Warehouse implementation only).

   Run SCM OWS jobs (for SCM Warehouse implementation only).

Moving Data Into the OWE

You use ETL jobs to move data into the OWE from the OWS. The following is an overview of the steps required to bring data into the OWE:

1. Run the setup - OWE jobs.
2. Run common dimension jobs.
3. Some EPM warehouses require OWE data.

   For these warehouses see steps below in, 'Moving Data Into the MDW.'

Moving Data Into the MDW

There are three methods of bringing data into the MDW:

- Extracting data from the OWS and moving it into the MDW.

  This is the most common method and the majority of your data is moved into the MDW in this way.

- Extracting data from the OWE and moving it into the MDW.

  Certain EPM warehouses use this method, which brings enriched, business unit-based data into the MDW.

  For example, the Profitability data mart in the FMS Warehouses uses OWE data that is output from the Global Consolidations analytical application.

- Extracting source data directly from a PeopleSoft source system and moving it into the MDW.

  This method bypasses the OWS and is only used when large volumes of data must be extracted, such as data used for the Marketing data mart in the CRM Warehouse.

You use ETL jobs to move data into the MDW. The following is an overview of the steps required to bring data into the MDW:

1. Run Global Dimension Jobs for Campus Solutions Warehouse

   Run Global Dimension Jobs for CRM Warehouse

   Run Global Dimension Jobs for FMS Warehouse

   Run Global Dimension Jobs for HCM Warehouse

   Run Global Dimension Jobs for SCM Warehouse

2. Run Local Dimension Jobs for Campus Solutions Warehouse

   Run Local Dimension Jobs for CRM Warehouse

   Run Local Dimension Jobs for FMS Warehouse

   Run Local Dimension Jobs for HCM Warehouse

   Run Local Dimension Jobs for SCM Warehouse

3. Run CSW SKU Jobs

   Run CRM SKU Jobs

   Run FMS SKU Jobs

   Run HCM SKU Jobs

   Run SCM SKU Jobs

4. Run Global-OWE Jobs for CRM Warehouse

   Run Global-OWE Jobs for FMS Warehouse

   Run Global-OWE Jobs for HCM Warehouse

   Run Global-OWE Jobs for SCM Warehouse

5. Run CRM-OWE jobs

   Run FMS-OWE jobs

   Run HCM-OWE jobs

   Run SCM-OWE jobs

For more information on the jobs required to load data into the MDW for your EPM Warehouse, see your warehouse specific PeopleBook (for example, the *PeopleSoft Enterprise Campus Solutions Warehouse PeopleBook*).

See Also

PeopleSoft Enterprise Campus Solutions Warehouse 9.1 PeopleBook, "Running Campus Solutions Warehouse Implementation Jobs"

Chapter 11, "Setting Up DataStage for EPM," page 255

Chapter 13, "Running Initial Setup Jobs," page 275

Chapter 14, "Importing Source Business Units into EPM to Create Warehouse Business Units," page 283

Understanding IBM WebSphere DataStage

PeopleSoft has an original equipment manufacturer (OEM) agreement with IBM for its WebSphere DataStage ETL tool and bundles this offering with PeopleSoft EPM. The IBM WebSphere DataStage tool uses ETL jobs to target specific data from a PeopleSoft source database and migrate it to the OWS, OWE, and MDW tables. IBM WebSphere DataStage is comprised of a server tool and client tool, which are discussed in more detail below.

IBM WebSphere DataStage provides the following features:

- Graphical design tools for designing ETL maps (called *jobs*)
- Data extraction from a variety of data sources
- Data aggregation using SQL SELECT statements
- Data conversion using predefined or user-defined transformations and functions
- Data loading using predefined or user-defined jobs

IBM WebSphere DataStage Terminology

You should be familiar with these IBM WebSphere DataStage terms:

Term	*Definition*
Administrators	Administrators maintain and configure DataStage projects.
Aggregator Stages	Aggregator stages compute totals or other functions of sets of data.
Data Elements	Data elements specify the type of data in a column and how the data is converted.
Container Stages	Container stages group reusable stages and links in a job design.

Term	*Definition*
DataStage Package Installer	This tool enables you to install packaged DataStage jobs and plug-ins.
Hashed File	A hashed file groups one or more related files plus a file dictionary. DataStage creates hashed files when you run a job that creates hash files (these are delivered with PeopleSoft EPM). Hashed files are useful for storing data from tables from a remote database if they are queried frequently, for instance, as a lookup table.
Hashed File Stage	A hashed file stage extracts data from or loads data into a database containing hashed files. You can also use hashed file stages as lookups. PeopleSoft ETL jobs use hashed files as lookups.
Inter-process Stage	An inter-process stage allows you to run server jobs in parallel on a symmetric multiprocessing system.
Plug-in Stages	Plug-in stages perform processing that is not supported by the standard server job stage.
Sequential File Stage	A sequential file stage extracts data from or writes data to a text file.
Transform Function	A transform function takes one value and computes another value from it.
Transformer Stages	Transformer stages handle data, perform any conversions required, and pass data to another stage.
Job	A job is a collection of linked stages, data elements, and transforms that define how to extract, cleanse, transform, integrate, and load data into a target database. Jobs can either be server or mainframe jobs.
Job Sequence	Job sequence invokes and runs other jobs.
Join Stages	Join stages are mainframe processing stages or parallel job active stages that join two input sources.
Metadata	Metadata is data about data; for example, a table definition describing columns in which data is structured.

DataStage Server

The IBM WebSphere DataStage server enables you to schedule and run your ETL jobs:

DataStage Sever

Three components comprise the DataStage server:

- Repository

  The Repository stores all the information required for building and running an ETL job.

- DataStage Server

  The DataStage Server runs jobs that extract, transform, and load data into the warehouse.

- DataStage Package Installer

  The DataStage Package Installer installs packaged jobs and plug-ins.

DataStage Client

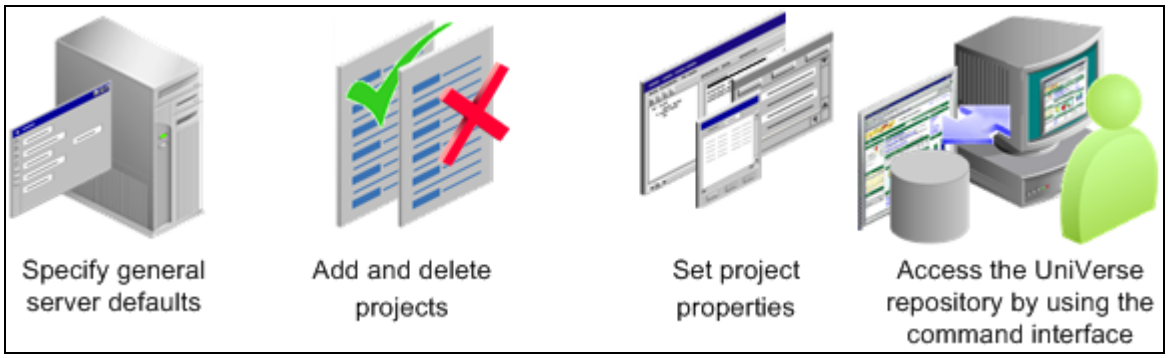
The IBM WebSphere DataStage client enables you to administer projects, edit repository contents, and create, edit, schedule, run, and monitor ETL jobs.

Three components comprise the DataStage client:

- DataStage Administrator
- DataStage Designer
- DataStage Director

DataStage Administrator

The DataStage Administrator enables you to:



DataStage Administrator

See Chapter 8, "Using DataStage Administrator," page 191.

DataStage Designer

DataStage Designer enables you to:

- Create, edit, and view objects in the metadata repository.
- Create, edit, and view data elements, table definitions, transforms, and routines.
- Import and export DataStage components, such as projects, jobs, and job components.
- Create ETL jobs, job sequences, containers, routines, and job templates.
- Create and use parameters within jobs.
- Insert and link stages into jobs.
- Set stage and job properties.
- Load and save table definitions.
- Save, compile, and run jobs.

See Chapter 9, "Using DataStage Designer," page 203.

DataStage Director

DataStage Director enables you to:

- Validate jobs.
- Schedule jobs.
- Run jobs.
- Monitor jobs.
- View log entries and job statistics.

See Chapter 10, "Using DataStage Director," page 245.

Key DataStage Components

IBM WebSphere DataStage contains many different components that support the ETL process. Some of these components include stages, jobs, and parameters. Only the following key DataStage components are discussed in this section:

- DSX Files
- Jobs
- Hashed Files
- Environmental Parameters
- Shared Containers
- Routines

A complete list of all DataStage components can be found in the *WebSphere DataStage Development: Designer Client Guide.*

DSX Files

PeopleSoft delivers a *.dsx file for each functional area within EPM. As part of your installation and configuration process you import the *.dsx file into a project that has been defined in your development environment. Included in the *.dsx file are various DataStage objects that define your project. The *.dsx files are organized by functional area and contain related ETL jobs.

To see a list of the PeopleSoft-delivered *.dsx files, refer to the file "DSX Files Import Description.xls" located in the following install CD directory path: <PSHOME>\SRC\ETL.

Each delivered *.dsx file contains the DataStage objects described in the following sections.

ETL Jobs

PeopleSoft delivers predefined ETL jobs for use with IBM WebSphere DataStage. ETL Jobs are a collection of linked stages, data elements, and transformations that define how to extract, transform, and load data into a target database. Stages are used to transform or aggregate data, and lookup information. More simply, ETL jobs extract data from source tables, process it, then write the data to target warehouse tables.

PeopleSoft deliver five types of jobs that perform different functions depending on the data being processed, and the warehouse layer in which it is being processed:

Load Stage	*Type*	*Description*
I	Source to OWS	Jobs in this category extract data from your transaction system and populate target warehouse tables in the OWS layer of the warehouse. Source to OWS jobs assign a source system ID (SRC_SYS_ID) for the transaction system from which you are extracting data and populate the target OWS tables with that ID.

Load Stage	Type	Description
I	Source to MDW	Jobs in this category extract data from your transaction system and populate target dimension and fact tables in the MDW layer of the warehouse. The Online Marketing data mart is the only product to use this type of job.
II	OWS to OWE	Jobs in this category extract data from the OWS tables and populate target D00, F00, and base tables in the OWE layer of the warehouse. OWS to OWE jobs perform lookup validations for the target OWE tables to ensure there are no information gaps and maintain referential integrity. Many of the jobs aggregate your transaction data for the target F00 tables.
II	OWS to MDW	Jobs in this category extract data from the OWS tables and populate target DIM and FACT tables in the MDW layer of the warehouse. OWS to MDW jobs generate a surrogate key that helps facilitate dimension key resolution. The surrogate key value is used as the primary key in the target DIM table and as the foreign key in the FACT table. The jobs also perform lookup validations for the target DIM and FACT tables to ensure there are no information gaps and maintain referential integrity.
II	OWE to MDW	Jobs in this category extract data from the OWE tables and populate target DIM and FACT tables in the MDW layer of the warehouse. Properties of this job type mirror those of the OWS to MDW job. OWE to MDW jobs generate a surrogate key that helps facilitate dimension key resolution. The surrogate key value is used as the primary key in the target DIM table and as the foreign key in the FACT table. The jobs also perform lookup validations for the target DIM and FACT tables to ensure there are no information gaps and maintain referential integrity.

All job types identified in the table are *incremental load* jobs. Incremental load jobs identify and extract only new or changed source records and bring it into target warehouse tables.

See Chapter 7, "Preparing to Load Source Data Into EPM," Understanding ETL Load Strategies in EPM, page 176.

ETL Jobs - Naming Convention

PeopleSoft use standard naming conventions for all ETL jobs; this ensures consistency across different projects. The following table provides the naming conventions for PeopleSoft delivered ETL jobs.

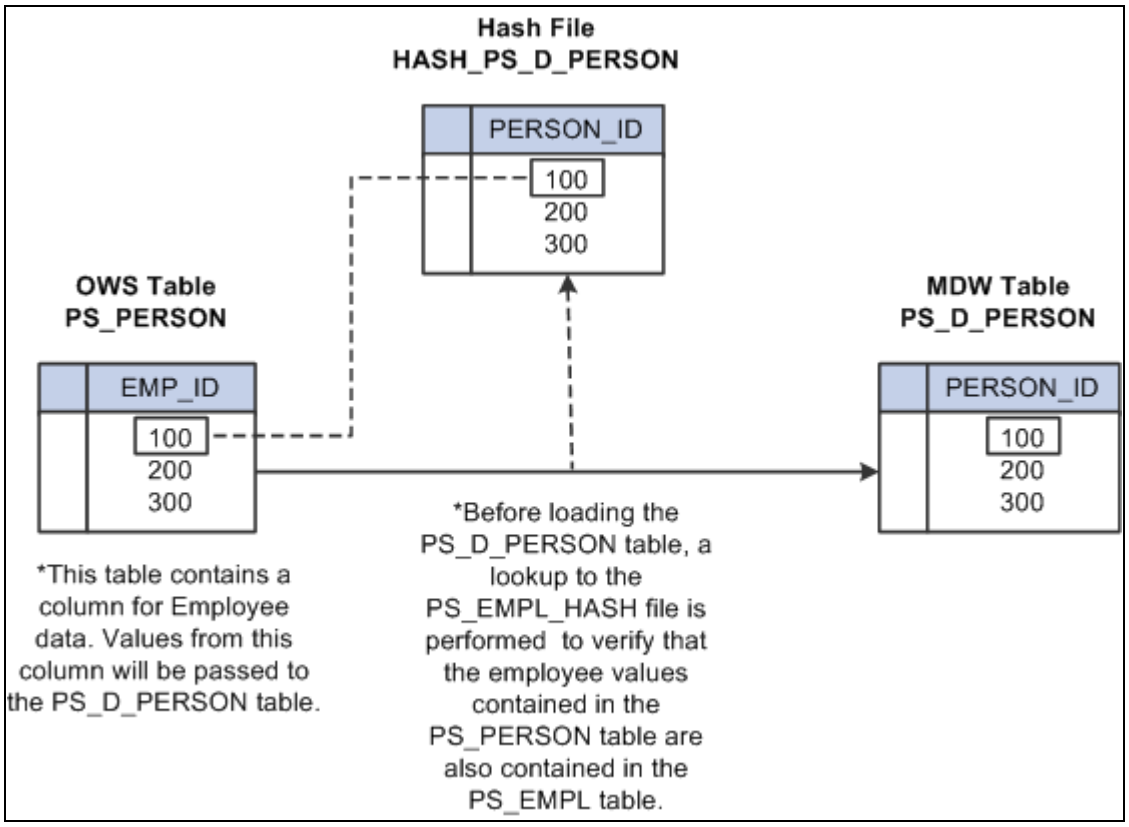
Object	Naming Convention	Example
Staging Server Job	J_Stage_[*Staging Table Name*]_[*Source Release*]_[*EPM Release*]	J_Stage_PS_AGING_TBL_FSCM91_EPM91

Object	*Naming Convention*	*Example*
Sequencer Job	SEQ_[*Staging Table Name*]_[*Source Release*]_[*EPM Release*]	SEQ_J_Stage_PS_AGING_TBL_FSCM91_EPM91
CRC Initial Load Job	J_Hash_PS_[*Staging Table Name*]_[*Source Release*]_[*EPM Release*]	J_Hash_PS_AGING_TBL_FSCM91_EPM91
Common Lookup Load Job	J_Hash_PS_[*Table Name*]	J_Hash_PS_D_LRNG_ENV
Dimension Job	J_Dim_PS_[*Dimension Table Name*]	J_Dim_PS_D_DEPT
Fact Job	J_Fact_PS_[*Fact Table Name*]	J_Fact_PS_F_ENRLMT
Enterprise D00 Job	J_D00_PS_[*D00 Table Name without D00 Suffix*]	J_D00_PS_ACCOMP_D00
Enterprise F00 Job	J_F00_PS_[*F00 Table Name without F00 Suffix*]	J_F00_PS_JOB_F00
OWE BASE Job	J_BASE_PS_[*Base OWE Table Name*]	J_BASE_PS_XYZ

Hashed Files

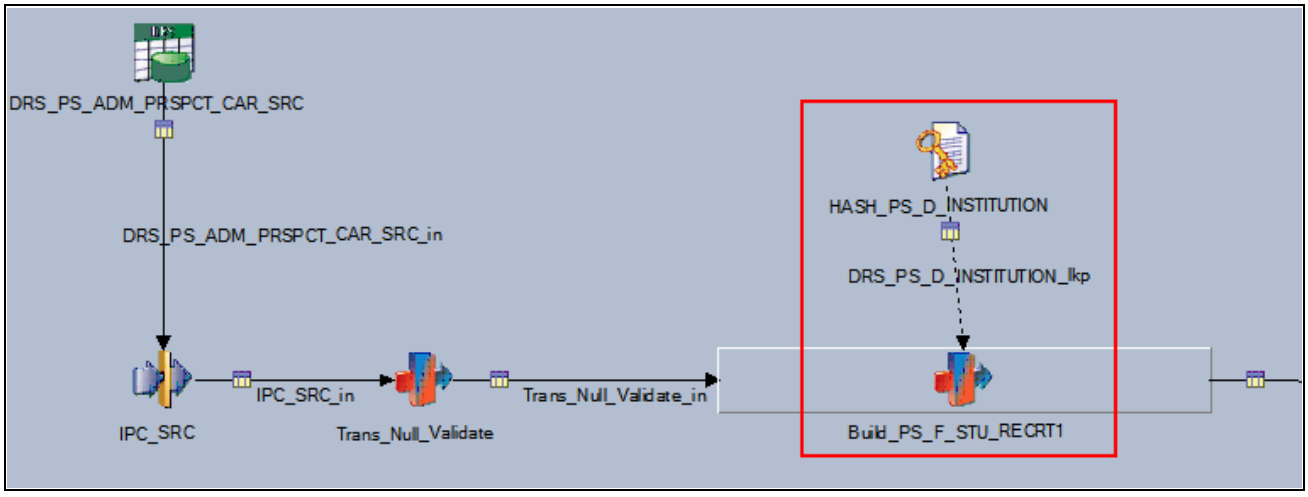
Hash files are views of specific EPM warehouse tables and contain only a subset of the data available in the warehouse tables. These streamlined versions of warehouse tables are used to perform data validation (lookups) within an ETL job and select specific data from lookup tables (such as sourceID fields in dimensions).

In the validation (lookup) process the smaller hash file is accessed, rather than the base warehouse table, improving performance. The following diagram provides an example of a hash file lookup in a job.



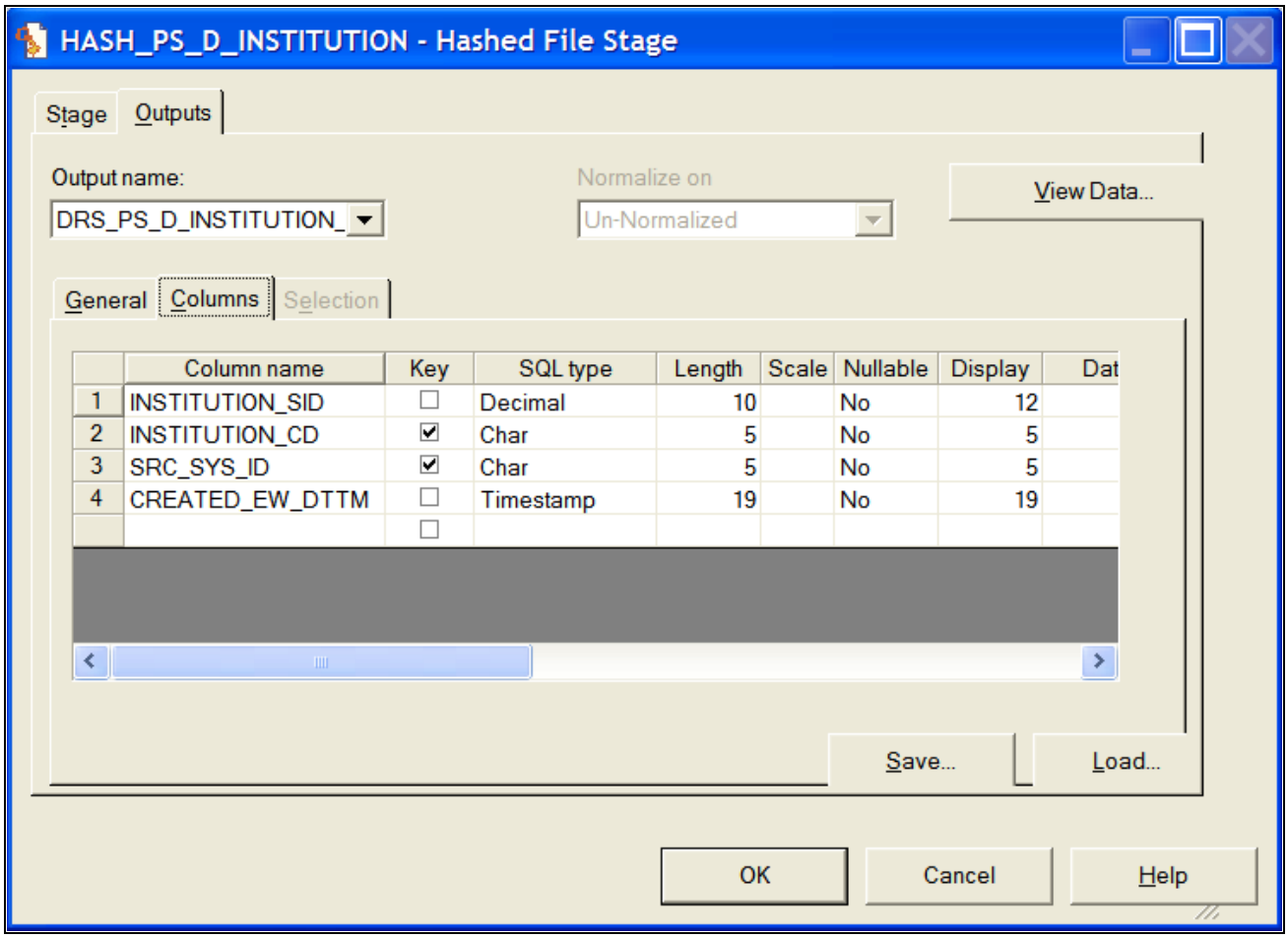
Lookup process using hash file

The following detailed view of an ETL job shows the Institution hashed file lookup in the Campus Solutions Warehouse J_Fact_PS_F_STU_RECRT job.



Institution hashed file lookup in the J_Fact_PS_F_STU_RECRT job

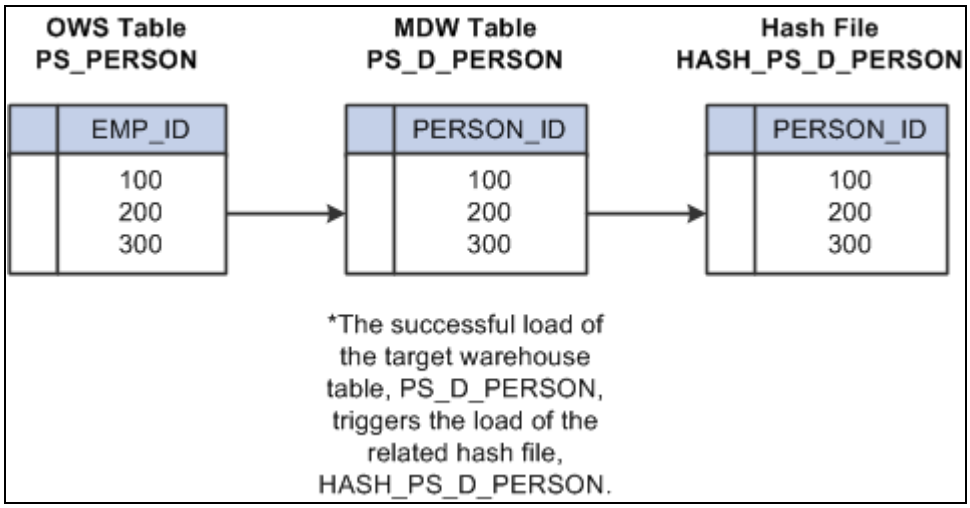
A detailed view of the hashed file stage reveals the fields (including keys) the lookup uses to validate Institution records.



Hashed file stage in the J_Fact_PS_F_STU_RECRT job

Because hash files are vital to the lookup process, jobs cannot function properly until all hash files are created and populated with data. Before you run any job that requires a hash file, you must first run all jobs that create and load the hash files—also called initial hash file load jobs.

After hash files are created and populated by the initial hash file load jobs, they are updated on a regular basis by the delivered sequencer jobs. Hash files are updated in the same job as its related target warehouse table is updated. In other words, both the target warehouse table and the related hash file are updated in the same sequencer job. The successful load of the target warehouse table in the job triggers the load of the related hash file. The following diagram provides an example of the this process.



Hash file update process

See Chapter 7, "Preparing to Load Source Data Into EPM," Understanding Data Validation and Error Handling in the ETL Process, page 179 and Chapter 7, "Preparing to Load Source Data Into EPM," Incremental Load and Update, Else Insert Logic, page 177.

Environmental Parameters

Environmental parameters are user-defined values that represent processing variables in your ETL jobs. Environmental parameters are reusable so they enable you to define a processing variable once and use it in several jobs. They also help standardize your jobs.

Though environmental parameters are reusable, PeopleSoft delivers specific environmental parameters for jobs related to each phase of data movement (such as the OWS to MDW jobs). Therefore, a single environmental parameter is not used across all ETL jobs, rather a subset of variables are used depending on the specific functionality of the job.

See Appendix C, "ETL Reference Documents," Environmental Parameters Information, page 671.

Shared Containers

Shared containers are reusable job elements. A shared container is usually comprised of groups of stages and links, and is stored in the DataStage repository. You can use shared containers to make common job components available throughout your project. Because shared containers are reusable you can define them once and use them in any number of your ETL jobs. PeopleSoft delivers the following shared containers:

- StoreLangStagingList
- StoreMaxLastUpdDttm
- StoreMaxRecordID
- StorMaxSnapDate
- StoreMaxValueDecimal
- FactStoreMaxRecordID

Routines

Routines are a set of instructions, or logic, that perform a task within a job. For example, the *ToInteger* routine converts the input value to an integer. Because routines are reusable you can use them in any number of your ETL jobs.

See Appendix C, "ETL Reference Documents," Routine Descriptions, page 671.

IBM Documentation

For more details on the IBM WebSphere DataStage tool and how to use it, refer to the IBM documentation listed below. You can install PDF versions of the IBM books as part of the IBM WebSphere tools install.

The following table lists the IBM documentation and the information provided.

IBM Book	*Description*
IBM Information Server: Planning Installation and Configuration Guide	Provides planning information and complete installation instructions for IBM Information Server. Also includes information about troubleshooting, validating the installation, and configuring the system.
IBM Information Server: Administration Guide	Describes how suite administrators can manage user access to components and features of IBM Information Server. In addition, describes how suite administrators can create and manage views of logged events and scheduled tasks for all components.
WebSphere DataStage Administration: Administrator Client Guide	Describes the WebSphere DataStage Administrator client and describes how to perform setup, routine housekeeping, and administration of the WebSphere DataStage engine.
WebSphere DataStage Administration: Deployment Guide	Describes how to package and deploy WebSphere DataStage jobs and associated objects to assist in moving projects from development to production.
WebSphere DataStage Administration: Director Client Guide	Describes the WebSphere DataStage Director client and explains how to validate, schedule, run, and monitor WebSphere DataStage parallel jobs and server jobs.
WebSphere DataStage Administration: National Language Support Guide	Describes how to use the national language support (NLS) features that are available in WebSphere DataStage when NLS is installed.
WebSphere DataStage Development: Designer Client Guide	Describes the WebSphere DataStage Designer client and gives a general description of how to create, design, and develop a WebSphere DataStage application
WebSphere DataStage Development: Server Job Developer Guide	Describes the tools that build a server job, and supplies programming reference information

Understanding ETL Load Strategies in EPM

This section provides an overview of ETL load strategies in EPM and discusses:

- Incremental Load with the Date Time Stamp
- Incremental Load Using Cyclical Redundancy Check
- Incremental Load and Update, Else Insert Logic
- Special Load Requirements

Overview of ETL Load Strategies in EPM

PeopleSoft delivers ETL jobs that extract data from your source transaction system and load it into target OWE and MDW dimension and fact tables. These jobs employ an *incremental load* strategy, which uses built-in logic to identify and extract only new or changed source records. When an incremental load job reads data from a source, it uses the date time stamp to identify new or changed records. When an incremental load job writes data to a target, it updates records using update, else insert logic. The benefit of the incremental load process is increased efficiency and faster processing during the extract and load process.

There are two types of incremental load strategies employed in PeopleSoft ETL jobs:

- Incremental load with a date-time stamp
- Incremental load using Cyclical Redundancy Check (CRC) logic

Note. If this is the first time you are populating your target warehouse tables with data, the incremental jobs recognize that you have no existing data in your tables and perform a complete extract of your source records. Subsequent runs of the incremental jobs will extract only new or changed records.

Incremental Load with the Date Time Stamp

To ensure only new or changed records are extracted into EPM , a date time stamp is associated with each record in the EPM tables. Target warehouse tables have a DTTM column which holds current date time values for each record, but a separate hash file stores the last update date time stamp.

When an incremental load job reads a table, it uses the built-in filter condition, [DTTM_Column] > [%DateTimeIn('#LastModifiedDateTime#')], to determine whether any records in the table are new or changed since the last load. The last update date time is retrieved from the related hash file using the *GetLastUpdDateTime* routine. If the retrieved date time is less than the current value in the DTTM column, the record will be updated in the EPM table. This process can be done quickly because the DTTM column is the only value being processed for each record.

Each time a new or updated record is loaded, the present date time stamp is recorded for the last update time stamp and is used as a basis for comparison the next time the incremental load job is run.

Note. If the last update time field is null for a record, the record is processed each time the job is executed.

Incremental Load Using Cyclical Redundancy Check

Source table records might not have a date time stamp field. When source table records lack a date time stamp, a *cyclical redundancy check (CRC)* must be performed to determine new or changed records. Unlike the traditional date time lookup process which targets the DTTM column for each record, the CRC process reads the entire record for each record in the source table and generates a CRC value to compare against the target warehouse record.

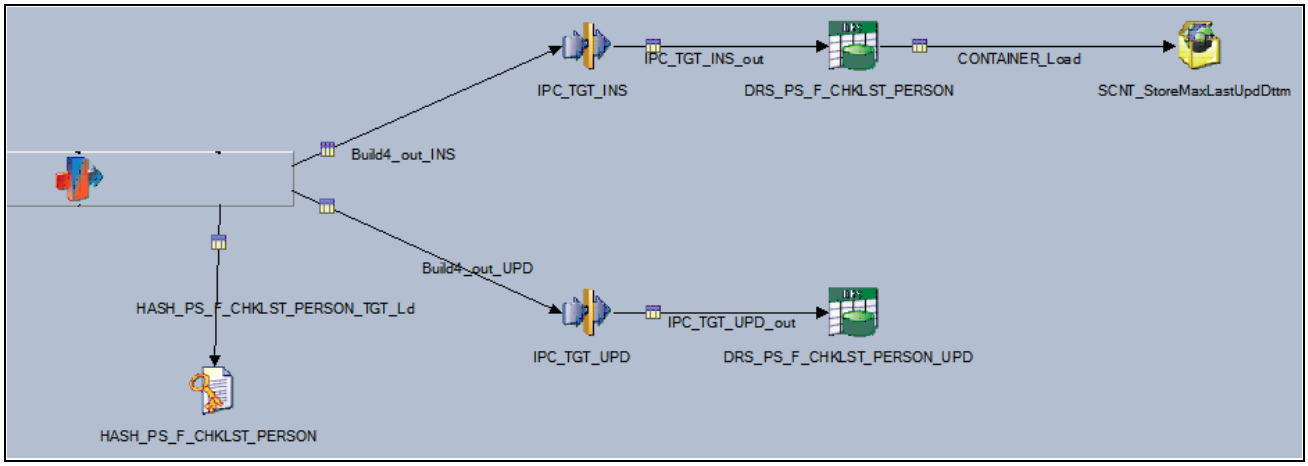
Incremental Load and Update, Else Insert Logic

If a record must be updated in an EPM table, the incremental load job uses update, else insert logic to determine how to load the new record. The update, else insert process works by using a hashed file lookup to compare all the key values coming in from the source table record against the same key values in the target table.

If the same key values exist in the EPM target table, the associated record is overwritten with the new values coming from the source.

If the key-comparison process finds that the key values do not exist in the EPM target table (it is a new record), a new record is inserted into the target table.

The following detailed view of an ETL job demonstrates the update, else insert process, including the related hashed file lookup:



ETL update, else insert process

Note. For OWE D00 and MDW dimension tables, only type 1 slowly changing dimension is supported; history is not maintained.

Special Load Requirements

The complex process behind integrating and aggregating disparate source data can create some special load requirements in EPM. For example, subrecords are used extensively in EPM target tables to provide additional depth and breadth of processing.

Passing Default Values to EPM Target Tables

Due to data aggregation and other processing requirements, EPM target tables may contain columns that do not exist in your source transaction tables. Because of the differences between source and EPM columns, there are sometimes no source values to populate the EPM columns. Therefore, default values must be used to populate the EPM columns instead.

Warehouse Layer	Data Type	Default Value
OWS	Char	'-'
	Num	0
	Date	Null
OWE	Char	' '
	Num	0
	Date	Null
MDW	Char	'-'
	Num	0
	Date	Null

For MDW fact records, fact rows coming from the source normally contain a valid reference to an existing row in the dimension table, in the form of a foreign key using a business key field. However, occasionally a fact row does not contain the dimension key. To resolve this issue, each MDW dimension contains a row for *Value Not Specified*, with predefined key values of zero—for a missing numeric value—and a hyphen—for a missing character value.

PeopleSoft delivers several routines to pass default values to the EPM columns. For example, the routine *GetNumDefault* is used to pass numeric default values to a target warehouse table. A separate routine is delivered for each data type (such as varchar and numeric).

Target Table Subrecords

Subrecords are a collection of specific columns that repeat across multiple EPM target tables. Subrecords can perform a variety of functions, including tracking data to its original source and facilitating customizations that enable type 2 slowly changing dimensions. For example, the subrecord *LOAD_OWS_SBR* contains columns such as CREATED_EW_DTTM, LAST_UPD_DTTM, and BATCH_SID which help track target warehouse table load history.

It is important to populate subrecords with the appropriate data. Thus, it is important that you thoroughly familiarize yourself with the PeopleSoft delivered subrecords and their associated columns.

Understanding Data Validation and Error Handling in the ETL Process

Accurate reporting is completely dependent on the data stored in EPM data warehouse OWE and MDW tables; if incomplete or incorrect data resides in these tables, reporting and analysis will be flawed, and essentially worthless. Given the considerable dependence on warehouse data, all source data entering EPM must be validated.

Typically source transaction systems have their own mechanisms to enforce data integrity, including database schema declarative referential integrity, runtime checks (such as database triggers), or application based checks. PeopleSoft Enterprise source systems implement data consistency checks in the application layer. More specifically, data consistency is enforced when data is entered using PIA pages. However, even with these source validations in place, one should not assume data integrity. For example, it is possible to enter data into a PeopleSoft Enterprise source system using a back-end mechanism, bypassing standard data validation checks. Hence, source data must be re-validated upon entry into EPM.

PeopleSoft EPM provides data validation that ensures:

- Each row in a fact table (*_F00 or F_*) has corresponding dimension and fact values.
- Source business unit and setID are properly mapped to EPM warehouse business unit and setID values.
- Source codes are properly mapped to EPM values.

EPM data validations are performed when you run ETL jobs; specific OWS jobs contain embedded logic that automatically perform the validations when you run the job. More specifically, data validation is performed in the job using lookup validations. Like other lookups, the validation process uses hashed files to lookup data and verify the integrity and completeness of the data. Embedded lookup validations can perform dimension key validation (for example, verifying that customer ID fact value has a corresponding customer ID dimension value) and general key validation (for example, verifying the pre-fact customer ID in the OWS table has a corresponding customer ID in the OWE or MDW table).

Because we want to ensure that complete, accurate data resides in the OWE and MDW layers, data validations are embedded in the jobs that load data from the OWS to the OWE and MDW. Therefore, data that passes the validation process is loaded into OWE and MDW target tables, while data that fails the validation process is redirected to separate error tables in the OWS. This ensures that flawed data never finds its way into the target OWE and MDW tables.

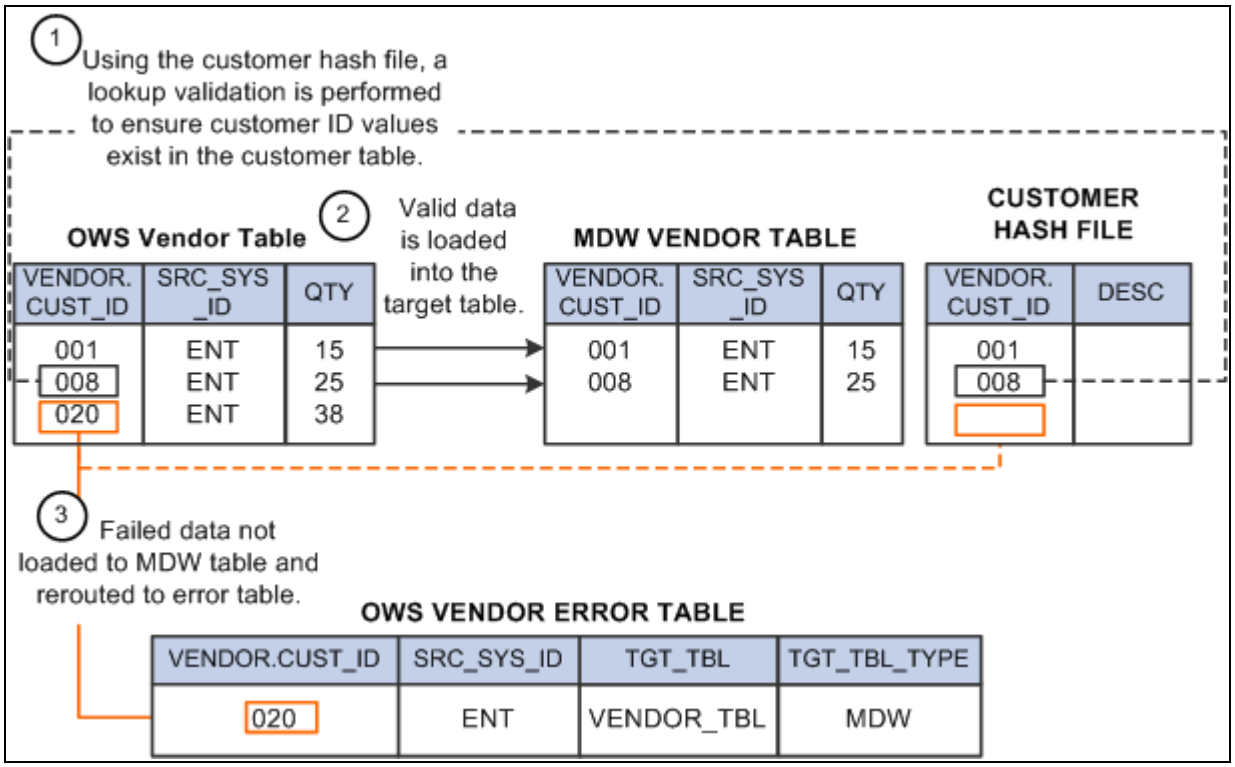
Error tables log the source values failing validation to aid correction of the data in the source system. There is an error table for each OWS driver table. OWS driver tables are those tables that contain the primary information for the target entity (for example customer ID). After flawed data is moved to the error table you must review this information and correct the problem in the source system.

See Chapter 2, "Understanding PeopleSoft Enterprise Performance Management," OWS Error Tables, page 11.

Note. PeopleSoft does not provide an error correction mechanism to fix data failing validation. Failed data should be corrected in the source system.

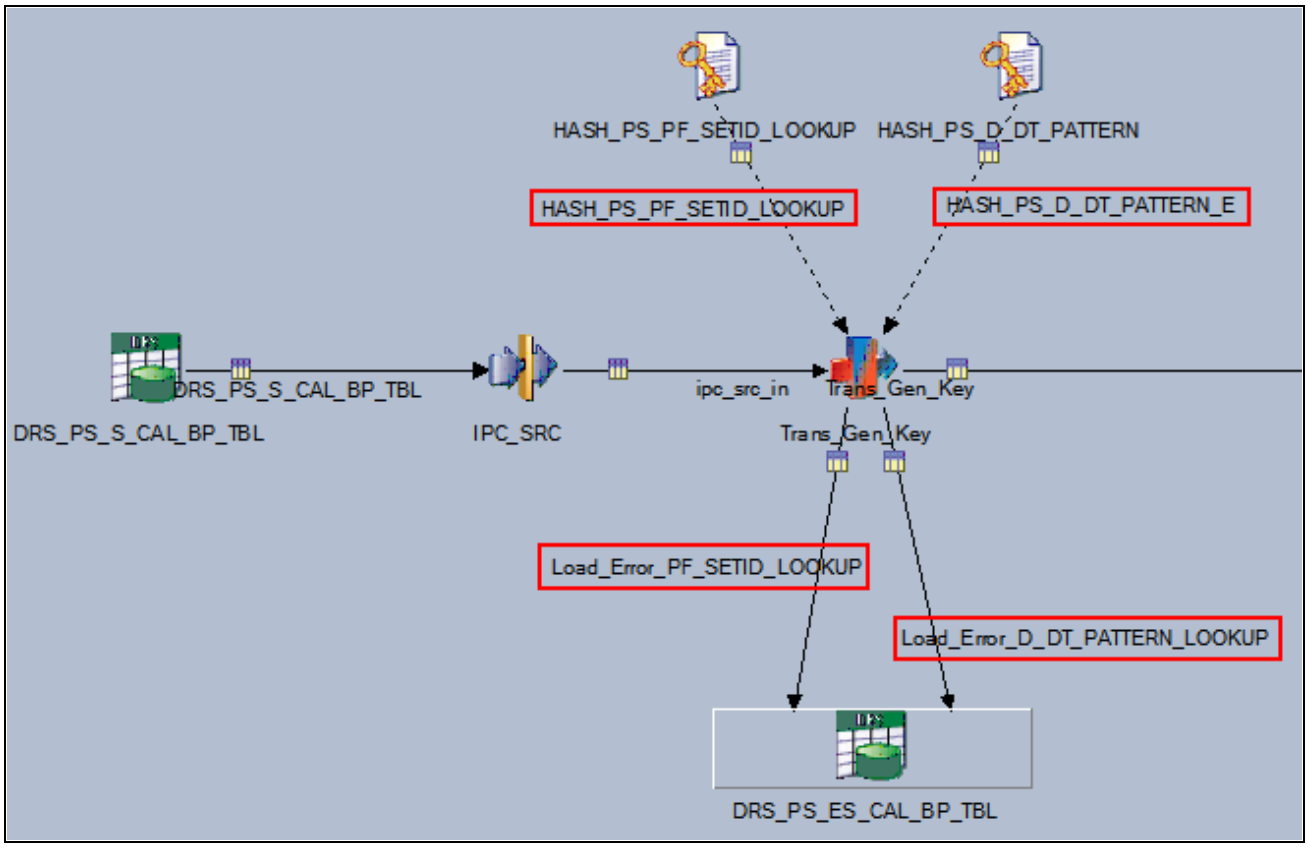
Example

The following graphic demonstrates a hypothetical ETL lookup process and represents a typical data validation process.



Sample ETL data validation process

The following graphic represents the data validation process in the PeopleSoft delivered J_DIM_PS_D_DET_BUDGET job:



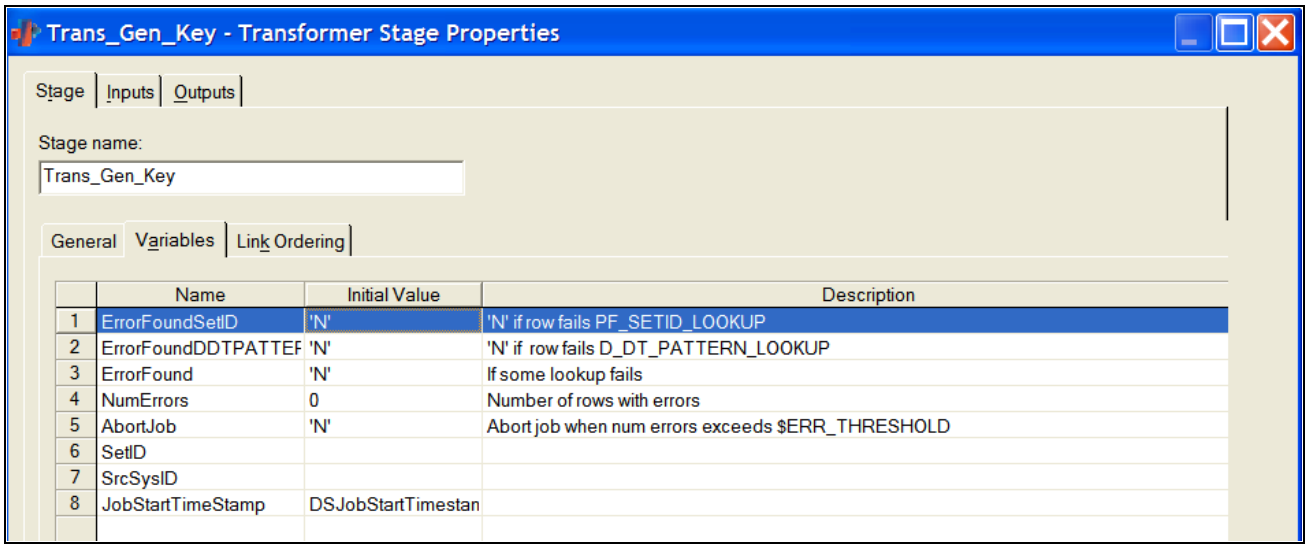
Data Validation in the J_DIM_PS_D_DET_BUDGET Job

Note that two hashed file validations are performed on the source data: the HASH_PS_PF_SETID_LOOKUP (which validates SETID) and HASH_PS_D_DT_PATTERN (which validates pattern code). Any data failing validation of these lookups is sent to the OWS error table (PS_ES_CAL_BP_TBL) via the Load_Error_PF_SETID_LOOKUP and Load_Error_D_DT_PATTERN_LOOKUP.

A closer look at the stage variables in the Trans_Gen_Key transformer stage demonstrate how the data validation process works:

Stage Variables	
Derivation	**Stage Variable**
If HASH_PS_PF_SETID_LOOKUP.NOTFOUND Then 'Y' Else 'N'	ErrorFoundSetID
IF HASH_PS_D_DT_PATTERN_E.NOTFOUND THEN 'Y' ELSE 'N'	ErrorFoundDDTPATTERN
If $ERR_VALIDATE = 'Y' AND (ErrorFoundSetID = 'Y' OR ErrorFoundDDTPATTERN = 'Y') Then 'Y' Else 'N'	ErrorFound
If ErrorFound = 'Y' Then NumErrors+1 Else NumErrors	NumErrors
If NumErrors > $ERR_THRESHOLD Then UtilityAbortToLog("Number of Errors exceeded the threshold limit a	AbortJob
NullToCharDefault(ipc_src_in.SETID)	SetID
NullToCharDefault(ipc_src_in.SRC_SYS_ID)	SrcSysID
	JobStartTimeStamp

Stage Variables in the Trans_Gen_Key Transformer Stage, 1 of 2



Stage Variables in the Trans_Gen_Key Transformer Stage, 2 of 2

Note that the *ErrorFoundSetID* and *ErrorFoundDDTPATTERN* stage variable derivations are set to *Y* if the SETID lookup or pattern code validations fail. The value of the *ErrorFound* stage variable, however, depends on the values of the two former stage variables, as well as the value of the *$ERR_VALIDATE* parameter, which can be configured to Y or N. If the $ERR_VALIDATE parameter is set to Y, rows that fail validation are written to the error table. If the value is set to N, rows that fail validation still pass to the target table.

Also note the *AbortJob* stage variable derivation uses the *$ERR_THRESHOLD* parameter to limit the number of error records allowed in the job. If the number of error records exceed the value set for the $ERR_THRESHOLD parameter, the job automatically aborts. For example, if $ERR_THRESHOLD is set to 50, the job aborts if the number of records with errors exceeds 50. You can set the value of the $ERR_THRESHOLD parameter to meet your specific business requirements.

Using the SetID lookup validation as an example, if a record fails validation, a Y value is assigned to the ErrorFoundSetID stage variable. If the $ERR_VALIDATE parameter is also set to Y, the failed record is sent to the PS_ES_CAL_BP_TBL error table.

Load_Error_PF_SETID_LOOKUP	
Constraint: $ERR_VALIDATE = 'Y' AND ErrorFoundSetID = 'Y'	
Derivation	Column Name
KeyMgtGetNextValueConcurrent('ES_CAL_BP_TBL')	ERROR_SID
SetID	SETID
NullToCharDefault(ipc_src_in.CALENDAR_ID)	CALENDAR_ID
NullToCharDefault(ipc_src_in.BUDGET_PERIOD)	BUDGET_PERIOD
SrcSysID	SRC_SYS_ID
'D_DET_BUDGET'	TARGET_TABLE_NAME
'MDW'	TARGET_TABLE_TYPE
'S_CAL_BP_TBL.SRC_SYS_ID | "CAL_BP_TBL" | S_CAL_BP_TBL.SETID | "D_DET_BUDGET"'	SOURCE_COLUMN_LIS
'PF_SETID_LOOKUP'	LOOKUP_TABLE
'SRC_SYS_ID | SRC_RECNAME | SRC_SETID | PF_RECNAME'	LOOKUP_COL_LIST
'Y'	LOAD_ERROR
$DATA_ORIGIN	DATA_ORIGIN
JobStartTimeStamp	CREATED_EW_DTTM
JobStartTimeStamp	LASTUPD_EW_DTTM
BATCH_SID	BATCH_SID
SrcSysID : ' | ' : 'CAL_BP_TBL' : ' | ' : SetID : ' | ' : 'D_DET_BUDGET'	ERR_DATA_VAL_LIST

Output Constraint for the Load_Error_PF_SETID_LOOKUP

For records that pass validation, an N value is assigned to the ErrorFound stage variable and the records are sent to the target table.

Trans_Gen_Key	
Constraint: ErrorFound = 'N'	
Derivation	Column Name
ipc_src_in.SETID	SRC_SETID
NullToCharDefault(HASH_PS_PF_SETID_LOOKUP.SETID)	SETID
ipc_src_in.CALENDAR_ID	CALENDAR_ID
ipc_src_in.BUDGET_PERIOD	BUDGET_PERIOD
ipc_src_in.SRC_SYS_ID	SRC_SYS_ID
ipc_src_in.BEGIN_DT	BEGIN_DT
ipc_src_in.END_DT	END_DT
ipc_src_in.PERIOD_NAME	PERIOD_NAME
NullToCharDefault(HASH_PS_D_DT_PATTERN_E.DT_PATTERN_CD)	DT_PATTERN_CD
NullToSIDDefault(HASH_PS_D_DT_PATTERN_E.DT_PATTERN_SID)	DT_PATTERN_SID
NullToCharDefault(HASH_PS_D_DT_PATTERN_E.DT_PATTERN_DESCR)	DT_PATTERN_DESCR

Output Constraint for the ErrorFound Stage Variable

Disabling Data Validation

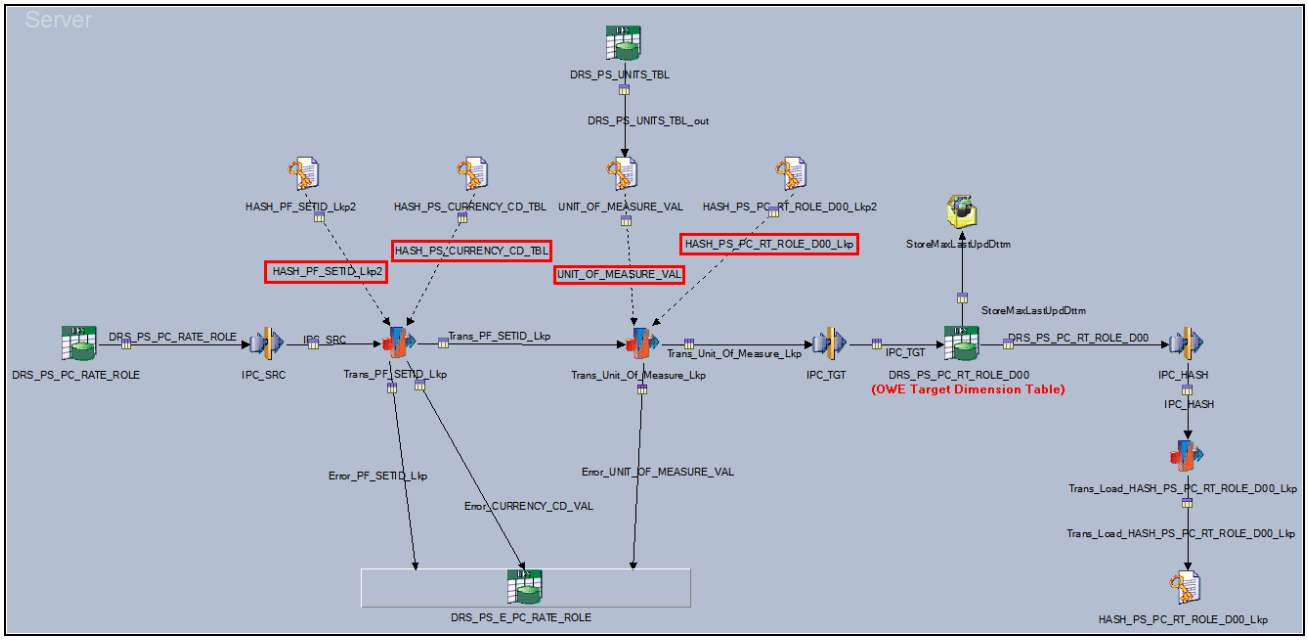
You can disable error validation in OWS jobs by configuring the value of the $ERR_VALIDATE parameter. By default the value is set to Y, which means that records failing validation are moved an error table. If you set the $ERR_VALIDATE value to N, records failing validation will still pass to the target table.

Understanding OWE Jobs

This section provides an overview of OWE fact and dimension load jobs.

OWE Dimension Load Jobs

A typical OWE dimension job loads data from an OWS source table (or in some cases, an OWE table) to a target OWE dimension table. The basic flow of an OWE dimension job starts with a DRS source stage and includes transformation stages to perform lookup validations against OWS, OWE, or MDW tables, depending on the job requirements.



Sample OWE Dimension Load Job (J_D00_PS_PC_RT_ROLE)

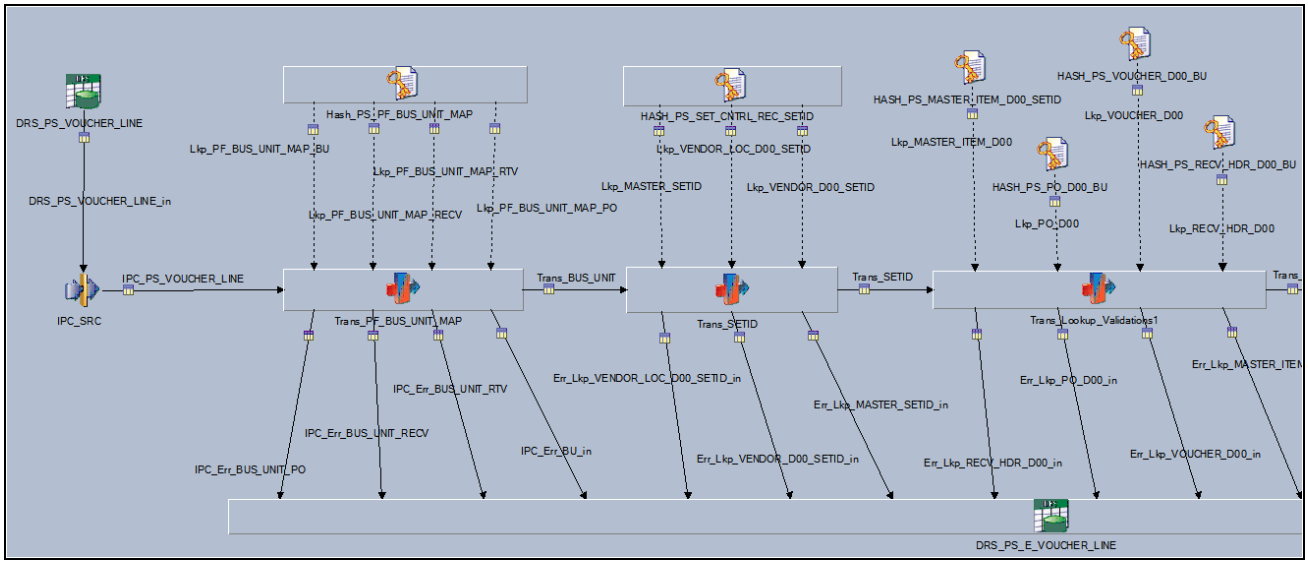
In this job, a SETID lookup (HASH_PF_SETID_Lkp) is performed since the target dimension table is SETID based. This validation verifies the existence of the value in the lookup table. If a record fails validation, it is inserted into the OWS error table (PS_E_PC_RATE_ROLE).

Currency Code and Unit of Measure validations are also performed in this job. Records failing these validations are sent to the PS_E_PC_RATE_ROLE error table.

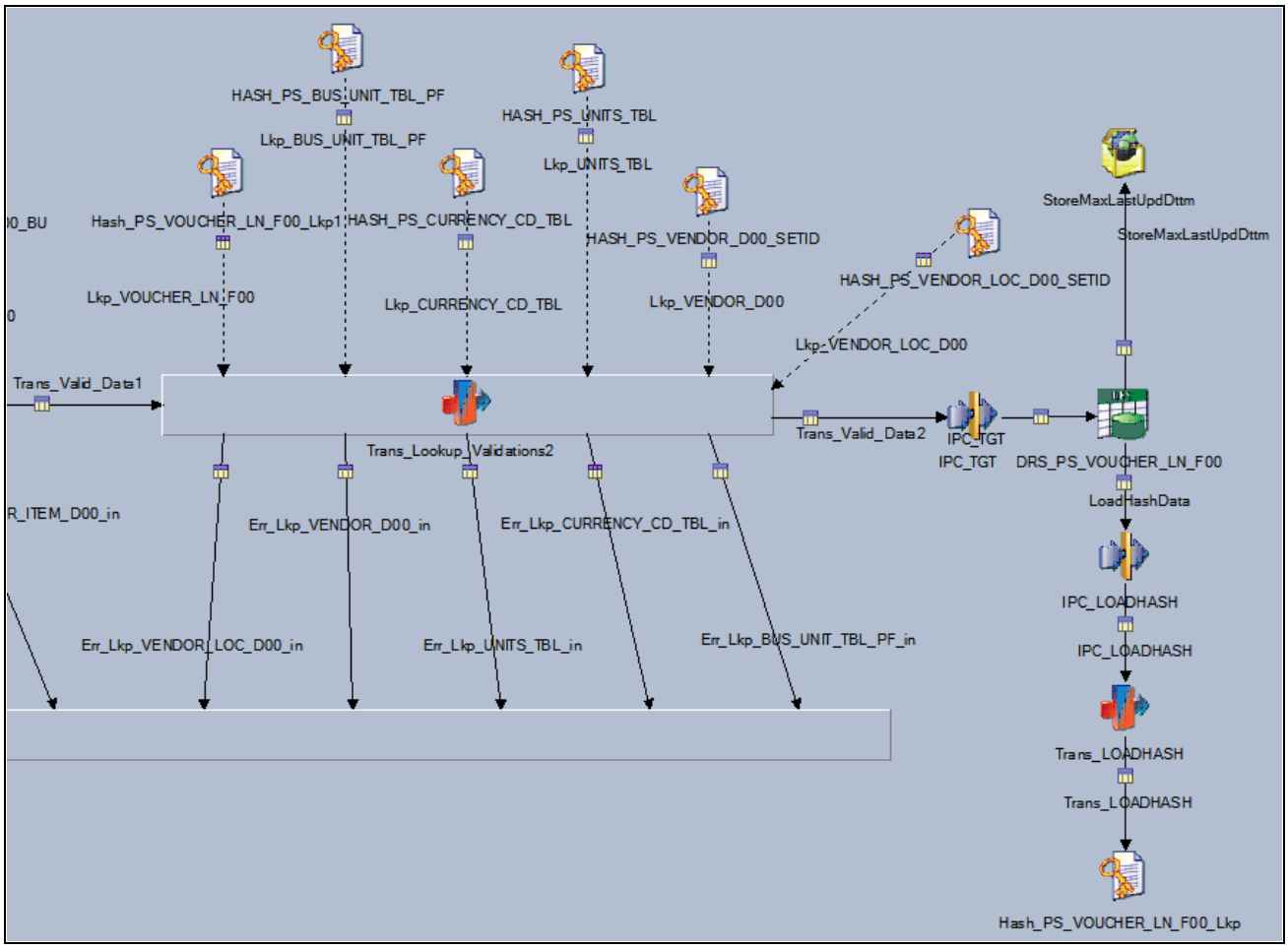
The HASH_PS_PC_RT_ROLE_D00 lookup is the final validation in this job and it is required for incremental loading of the OWE target dimension table (PS_PC_RT_ROLE_D00). This lookup fetches the CREATED_EW_DTTM value for records in the hashed file and determines whether equivalent business keys are already present. If a matching record exists in the hashed file, the same created date time is extracted from this lookup. The record is then loaded into the target DRS stage and the hash file used for incremental loading is updated.

OWE Fact Load Jobs

An OWE fact job loads data from an OWS source table (or in some cases, an OWE table) to a target OWE fact table. The basic flow of an OWE fact job starts with a DRS source stage and includes transformation stages to perform a lookup validations.



Sample OWE Fact Load Job (J_F00_PS_Voucher_LN), 1 of 2



Sample OWE Fact Load Job (J_F00_PS_Voucher_LN), 2 of 2

This fact job uses several lookup validations, including:

- Business unit validations using the HASH_PS_PF_BUS_UNIT_MAP hashed file
- SETID validations using the HASH_PS_SET_CNTRL_REC_SETID hashed file

  The SETID value is used in validations against OWE dimensions or other OWE fact tables.
- Currency code validations using the HASH_PS_CURRENCY_CD_TBL hashed file.

Records failing these validations are rerouted to the PS_E_VOUCHER_LN OWS error table.

Some OWE fact jobs also provide data transformation logic, such as aggregation of values or string manipulation.

The HASH_PS_VOUCHER_LN_F00 lookup is the final validation in this job and it is required for incremental loading of the OWE target fact table (PS_VOUCHER_LN_F00). This lookup fetches the CREATED_EW_DTTM value for records in the hashed file and determines whether equivalent business keys are already present. If a matching record exists in the hashed file, the same created date time is extracted from this lookup. The record is then loaded into the target DRS stage and the hash file used for incremental loading is updated. When a fact table is loaded destructively, the server job truncates the target table prior to loading data.

Understanding MDW Jobs

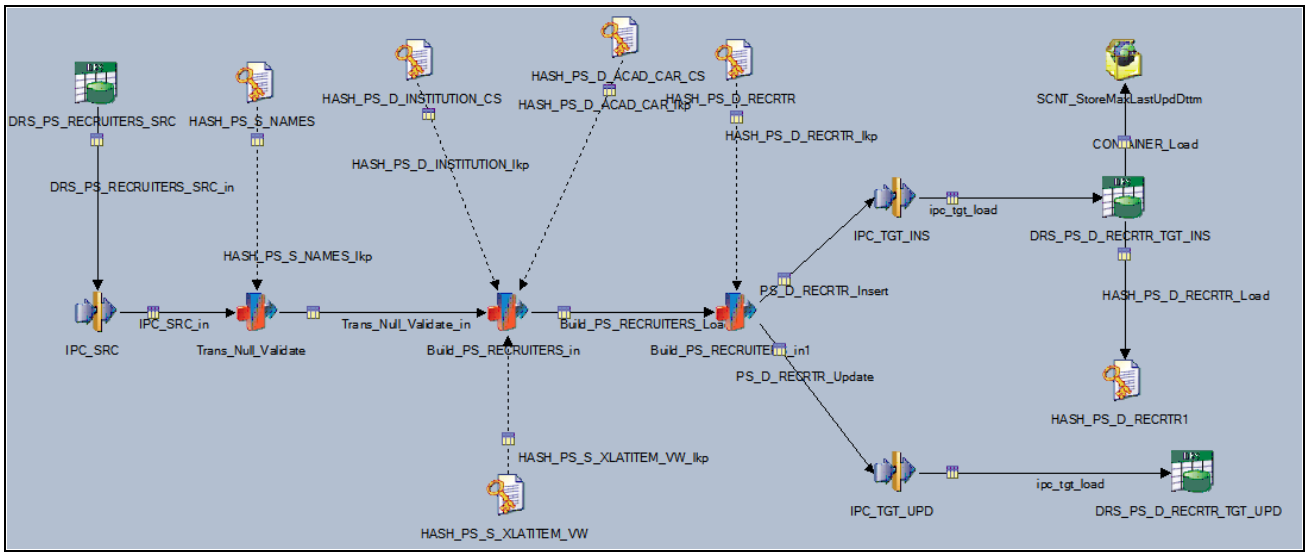
This section provides an overview of MDW fact and dimension load jobs.

MDW Dimension Load Jobs

A MDW dimension job loads data from an OWS source table (or in some cases, an OWE table) to a target MDW dimension table. The basic flow of a MDW dimension job starts with a DRS source stage and includes transformation stages with data validation lookups, when necessary, using the SID.

SID validations work in the same manner as the data validations described in the *Understanding Data Validation and Error Handling in the ETL Process* section above, except the SID is the unique key identifier used. The job will also contain lookups for attribute values, such as description fields.

The following is an example of a typical MDW dimension load job.

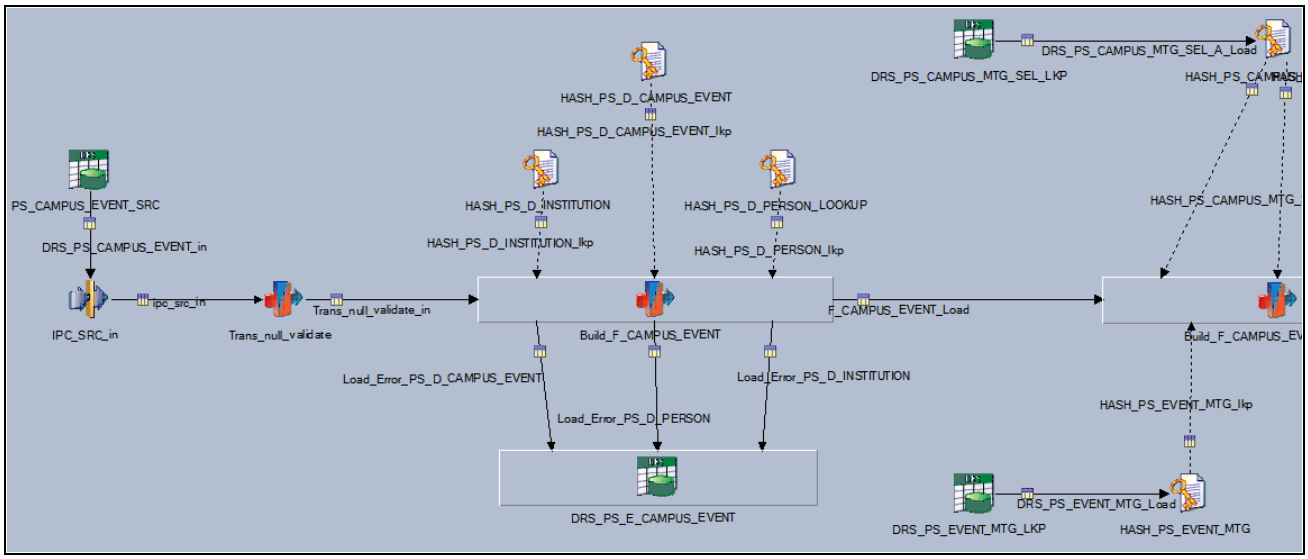


Sample MDW Dimension Load Job (J_DIM_PS_D_RECRTR)

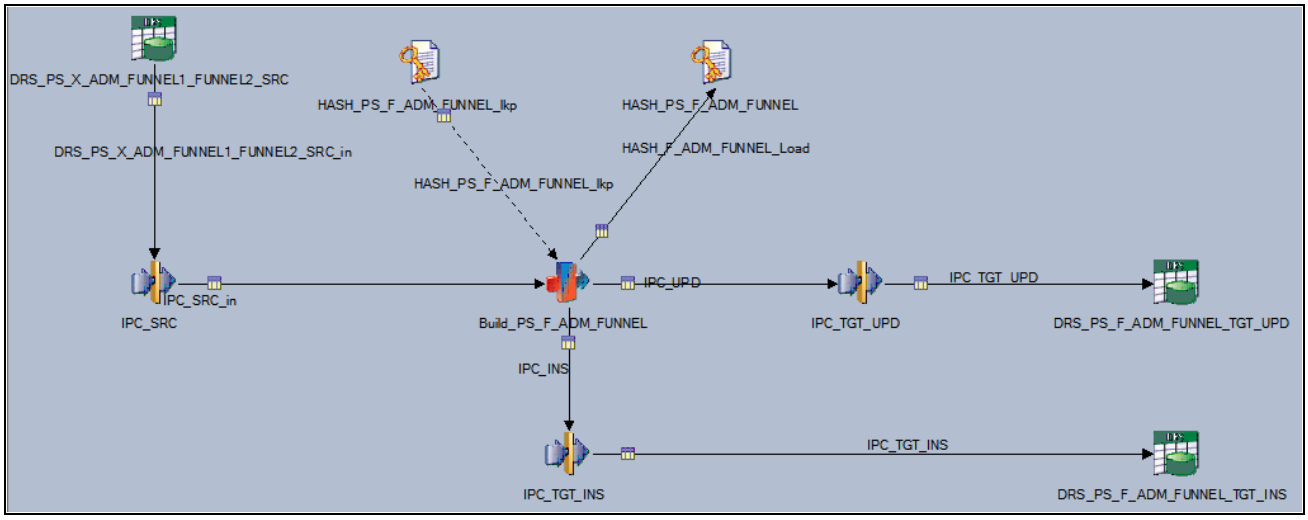
Next the job performs a lookup on the target dimension table hash file to check if equivalent business keys are already present for each record. If the record is present, the existing SID is used. If the record is not present, a new SID is generated. The job loads valid data into the target DRS stage and updates the hash file used for incremental loading.

MDW Fact Load Jobs

A MDW fact job loads data from an OWS source table (or in some cases, an OWE table) to a target MDW fact table. The basic flow of a MDW fact job starts with a DRS source stage and includes transformation stages to validate values for SID lookup dimension tables.

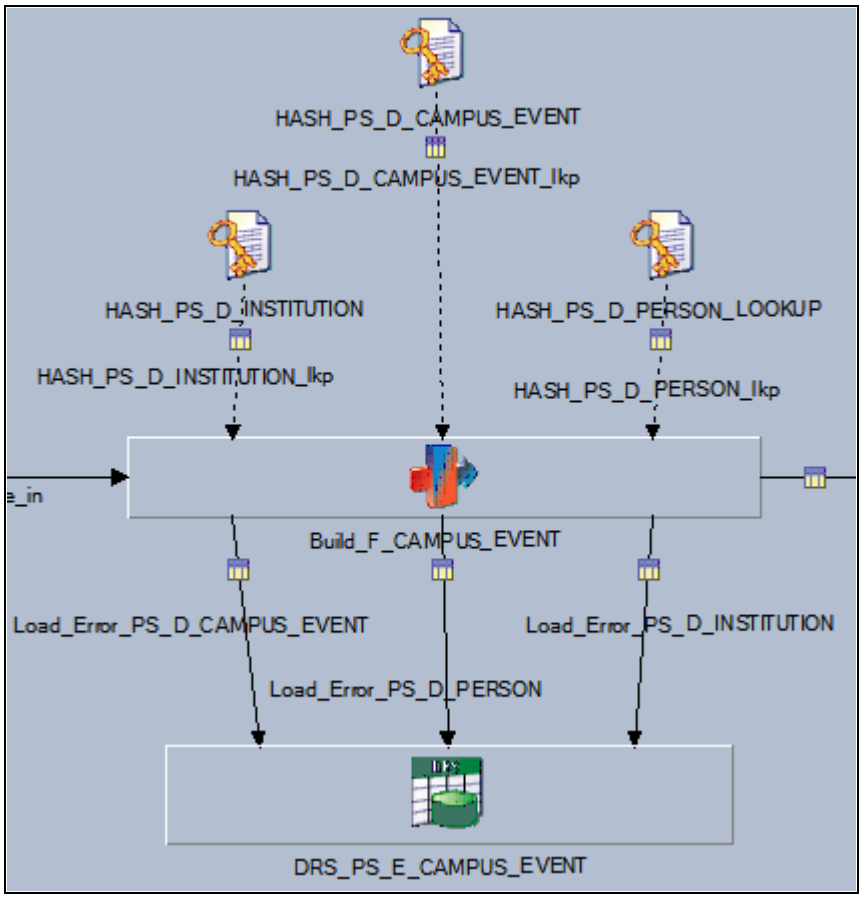


Sample MDW Fact Load Job (J_Fact_PS_F_CAMPUS_EVENT)



Sample MDW Fact Load Job (J_Fact_PS_F_ADM_FUNNEL)

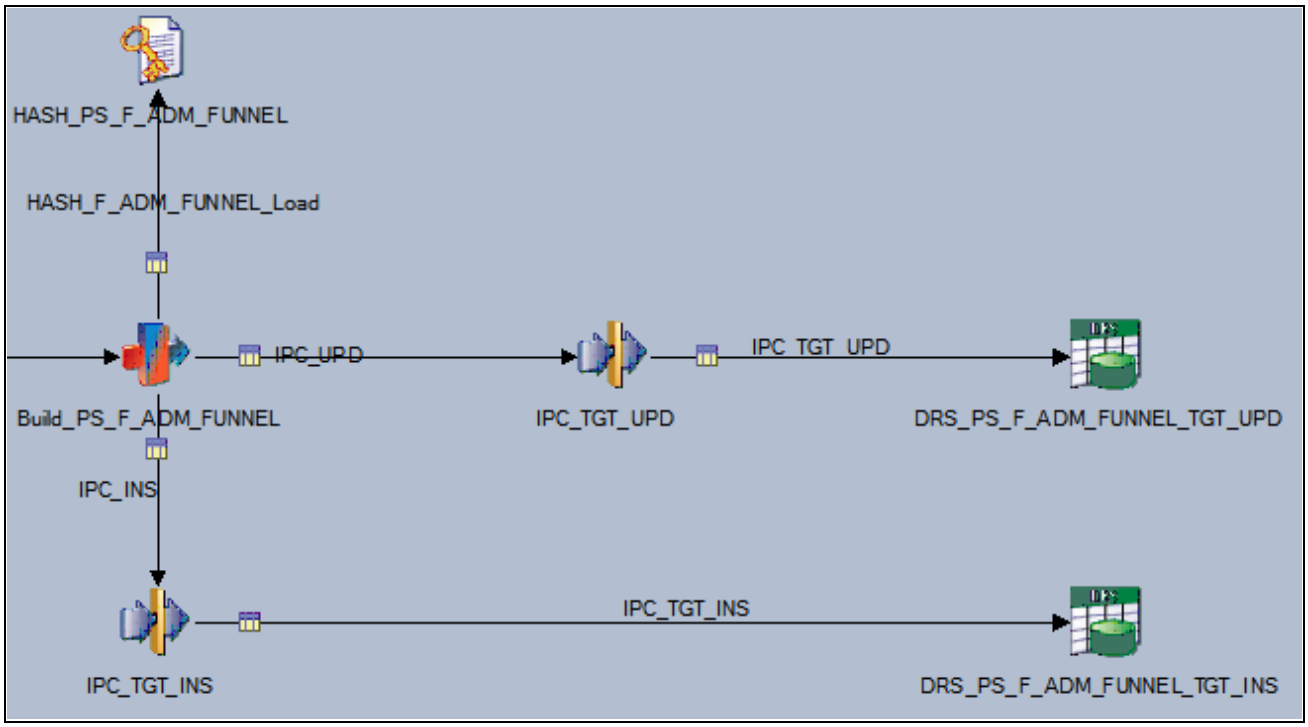
Because transaction tables are based on business unit and some dimension tables are SETID based, sometimes a *SETID indirection* lookup must be performed against the SETCTRL table to obtain the corresponding SETID for the business unit, and then use the value for the lookup. These lookups provide the values for the SID columns in the fact tables. The MDW fact job performs data validation lookups and diverts records that fail the lookup to an OWS error table (in this case, the PS_ECAMPUS_EVENT error table).



Data Validation and Error Handling in the J_Fact_PS_F_CAMPUS_EVENT Job.

Next, data transformations are sometimes performed in transformation stages, such as aggregation of values or string manipulation.

The HASH_PS_F_ADM_FUNNEL lookup is the final validation in this job and it is required for incremental loading of the MDW target fact table (PS_F_ADM_FUNNEL). This lookup fetches the CREATED_EW_DTTM value for records in the hashed file and determines whether equivalent business keys are already present. If a matching record exists in the hashed file, the same created date time is extracted from this lookup. The record is then updated in the target fact table. If the record is not present, a new record is inserted in the target fact table.



Update and Insert Data to Target Fact Table in the J_Fact_PS_F_ADM_FUNNEL Job.

The job also updates the hash file used for incremental loads. A very small number of MDW fact load jobs use destructive loading, in which case the server job truncates the target table prior to loading data.

Chapter 8

Using DataStage Administrator

DataStage Administrator enables you to specify general server defaults, administer projects, and set project properties.

The DataStage Administrator window is comprised of the General and Projects tabs.

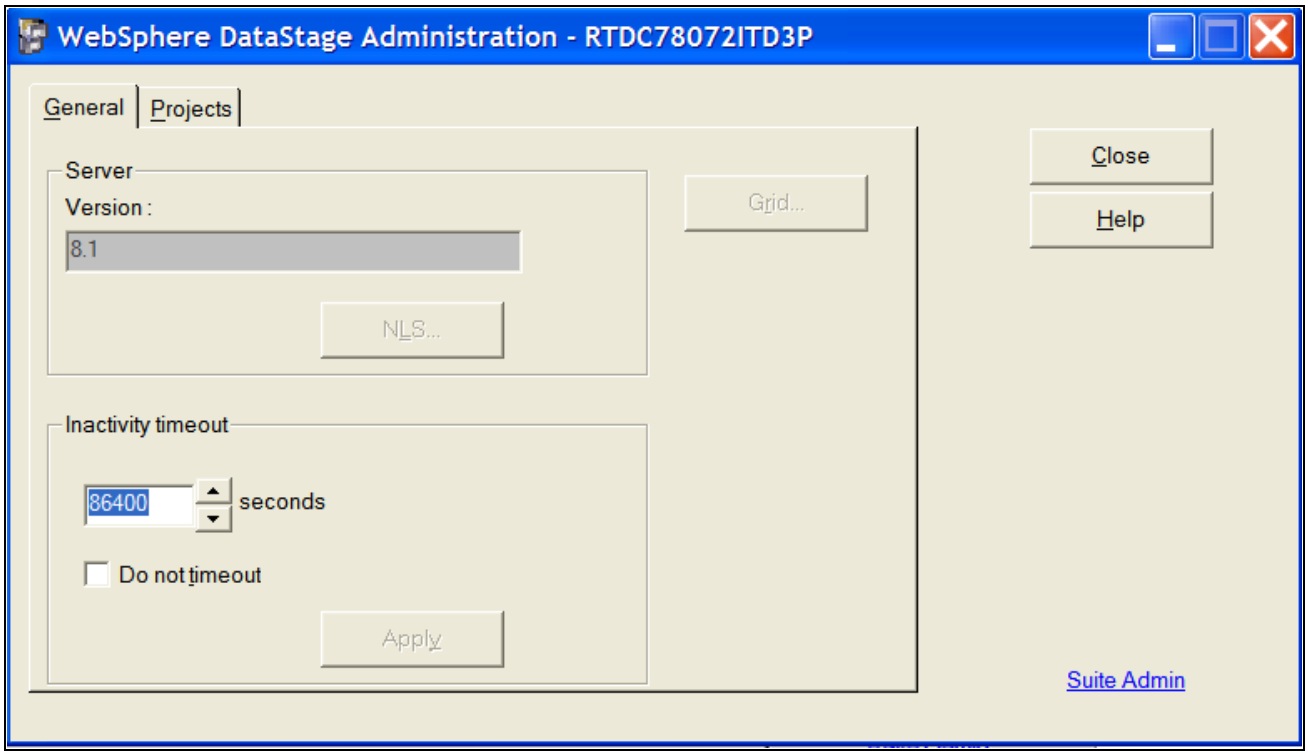
This chapter disscuses how to:

- Set DataStage Server properties.
- Set Project properties
- Set DataStage Server Licensing

Note. This chapter does not discuss all the features available for DataStage Administrator. For a complete view of DataStage Administrator functionality, please see the delivered IBM WebSphere documentation.

Setting DataStage Server Properties

Access the DataStage Administrator - General tab to set DataStage server properties.



DataStage Administrator - General Tab

You can change the following server-wide properties:

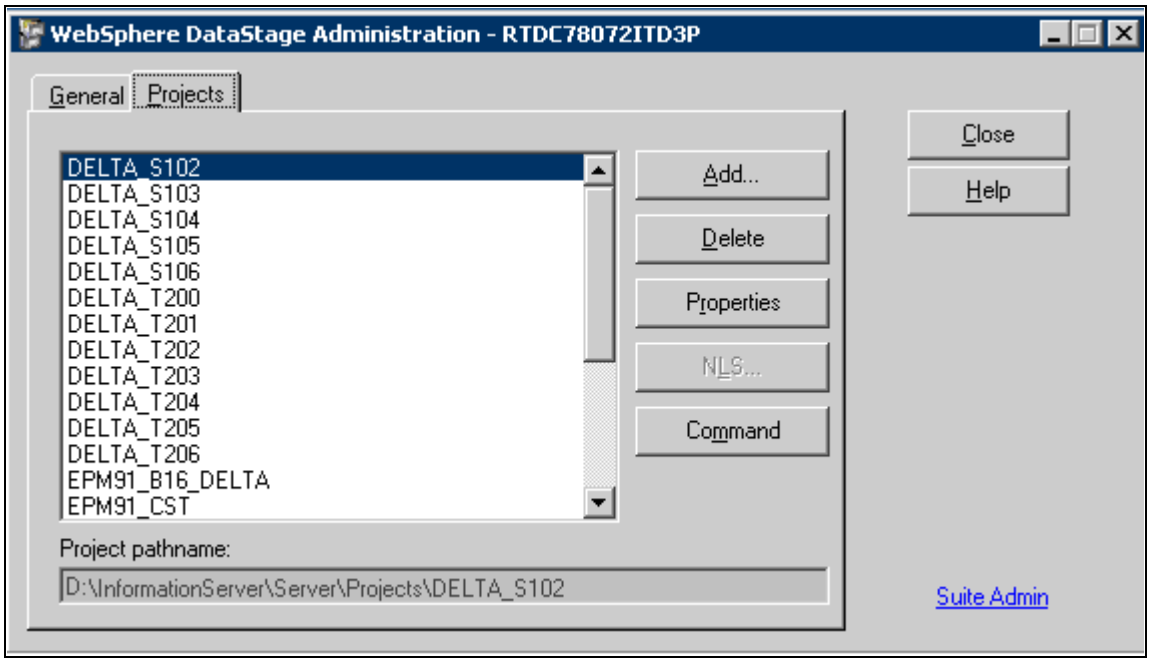
NLS Enable or disable National Language Support (NLS). DataStage supports the language you specify during the install without any further configuration. However, if your requirements change, you can reconfigure NLS to support different languages using DataStage Administrator. Note: You can only change the NLS character set in the DataStage Administrator. You enable and disable NLS support during install.

Inactivity Timeout Enter the number of seconds of inactivity allowed before the connection between the DataStage client and server times out.

Note. Server-wide property changes made by an administrator affect all projects on the server.

Setting Project Properties

Access the DataStage Administrator - Projects tab.



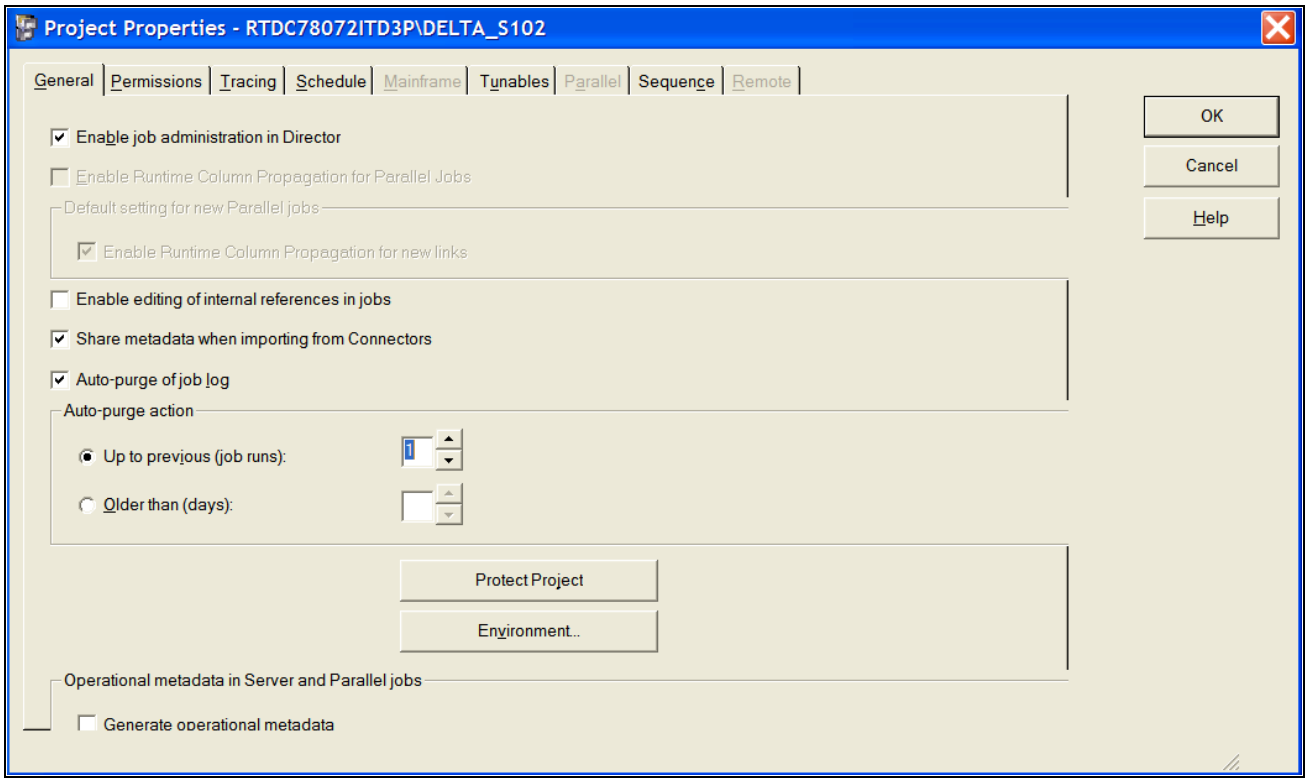
DataStage Administrator - Projects Tab

Using the DataStage Administrator - Projects tab, administrators can navigate to projects and:

- Add and delete projects.
- Set job administration options.
- Assign user access.
- Enable tracing on the server.
- Set up users for running scheduled jobs (Microsoft Windows only).
- Configure cache settings for hash file stages.
- Select compilation options when job sequences are created.

Project Properties - General Tab

Access the Project Properties - General tab (click the Properties button on the DataStage Administrator - Projects tab):



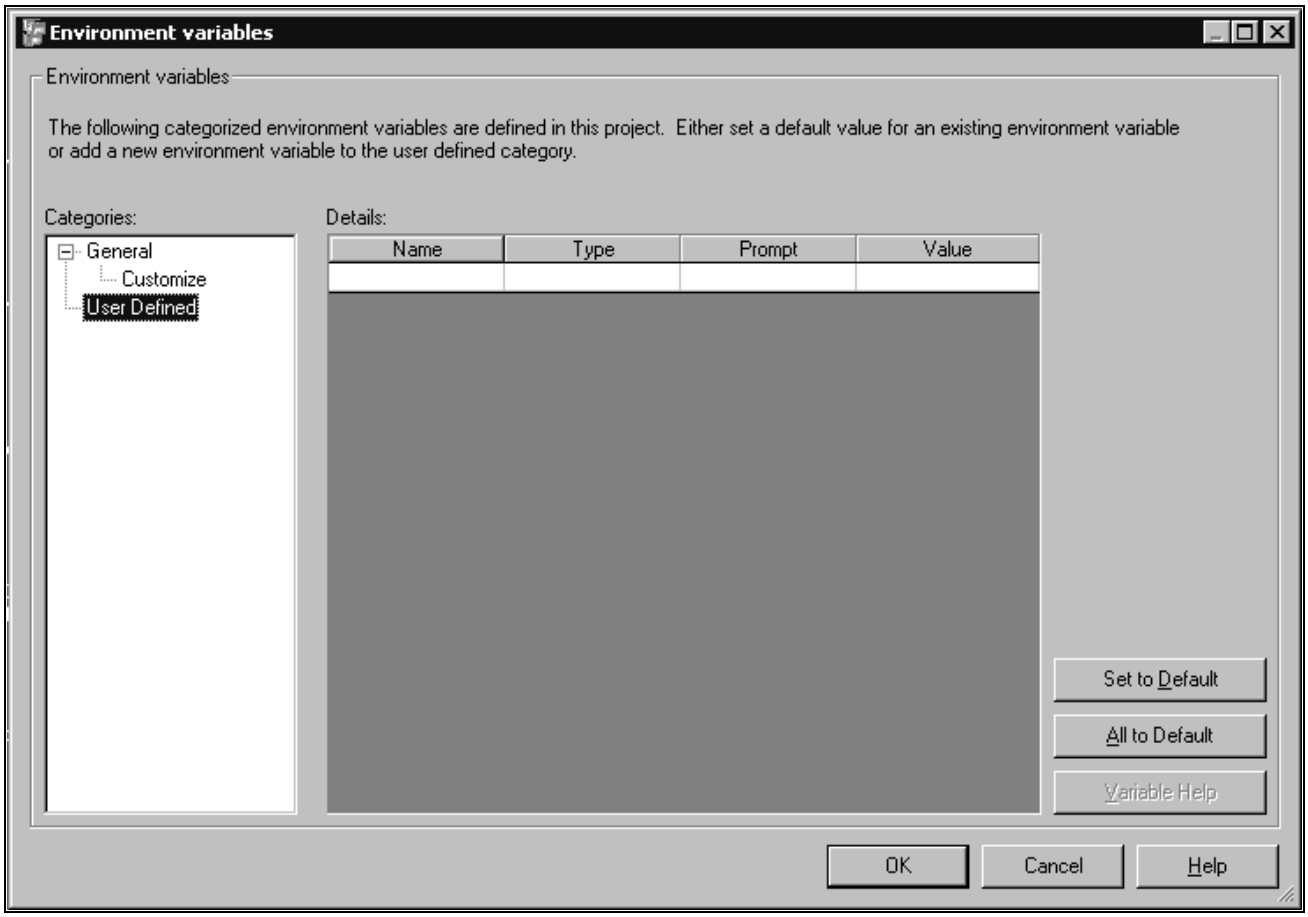
Project Properties - General Tab

The Project Properties - General tab includes the following options:

Enable job administration in Director	Select to use the Cleanup Resources and Clear Status File options from the Job menu of DataStage Director.
Enable Runtime Column Propagation in Parallel Jobs	If you have parallel jobs, select to enable stages to handle undefined columns during the job run. This setting propagates these columns throughout the rest of the job.
Enable remote execution of Parallel Jobs	Select to specify that parallel jobs in a project be deployed on USS systems.
Auto-purge of job log	Select to automatically delete the logs generated when you run a job, according to the criteria you select in the Auto purge action group box.
Up to previous (job runs) and **Over** (days old)	Select one of these options to delete jobs based on the number of job logs that you want to retain or based on the number of days old a job is. Enter the appropriate value in the adjacent field.
Protect Project	If you have Production Manager permissions, click to convert the project to a protected project to prevent its modification.
Generate Operational Metadata	Select this check box if you want parallel and server jobs in your project to generate operational metadata. You can override this setting in individual jobs if desired.

Setting Environment Variables

Click the Environment button on the Project Properties - General tab to set project-wide environmental variables.



Environment Variables Window

DataStage Administrator enables you to create user-defined environment variables and assign default values for existing variables used throughout a project.

Changing an environment variable affects all of the jobs in the project. To change an environment variable for each job, leave the Value column empty and specify the variable value in a job parameter instead. You can also override the value when the job runs.

To set a default value for an environment variable, select the variable type from the Environment Variable Tree in the left pane, and then enter a value in the right pane.

To create a new variable, select User Defined in the Environment Variable Tree, and then enter a new variable name, prompt, and value in the right pane.

Click *Set to Default* to set the selected variable to its installed default value.

Click *All to Default* to set all currently visible variables to their installed default values.

Click *Variable Help* to get information about the selected variable.

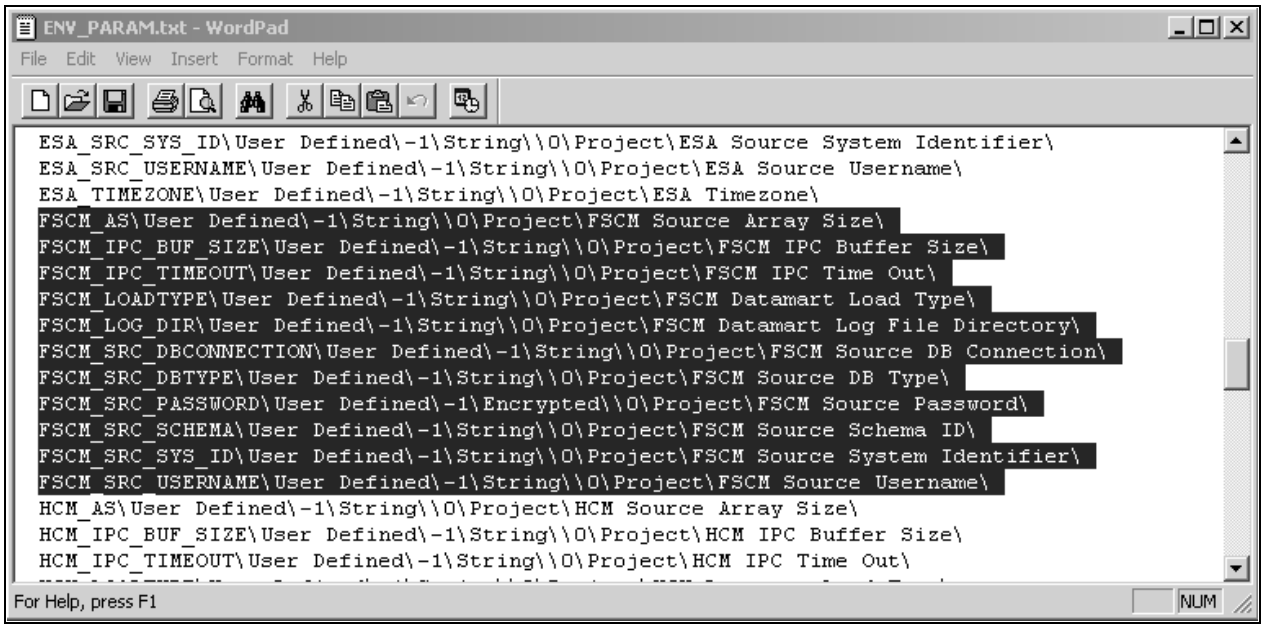
Setting Environment Variables - Example

To configure the delivered environment parameters:

1. Open DataStage Administrator and select your project.
2. Note the project path name of the selected project and close DataStage Administrator.
3. Use the project path to navigate to the DSPARAM file.

   The DSPARAM file should be located in that folder.
4. Open the DSPARAM file in Notepad.
5. Search for [EnvVarDefns].
6. Open the ENV_PARAM.txt file, and then select and copy the contents of the ENV_PARAM.txt file.

   You can copy specific entries based on the product.



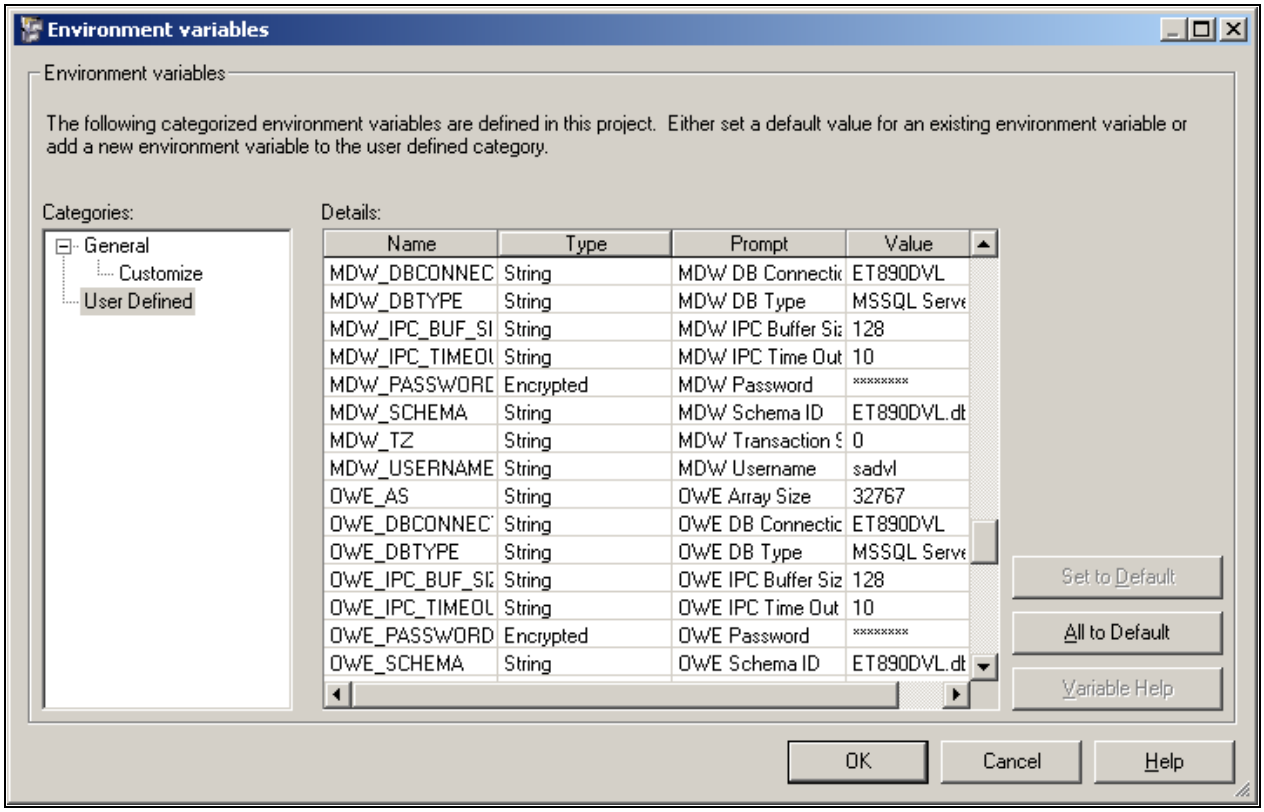
ENV_PARAM.txt File

7. Paste the copied contents to the DSPARAM file.

   The contents should be pasted below the line that contains the [EnvVarDefns] text.
8. Save the DSPARAM file.

9. Open DataStage Administrator, navigate to the Environmental Variables window, and select the User-Defined category.

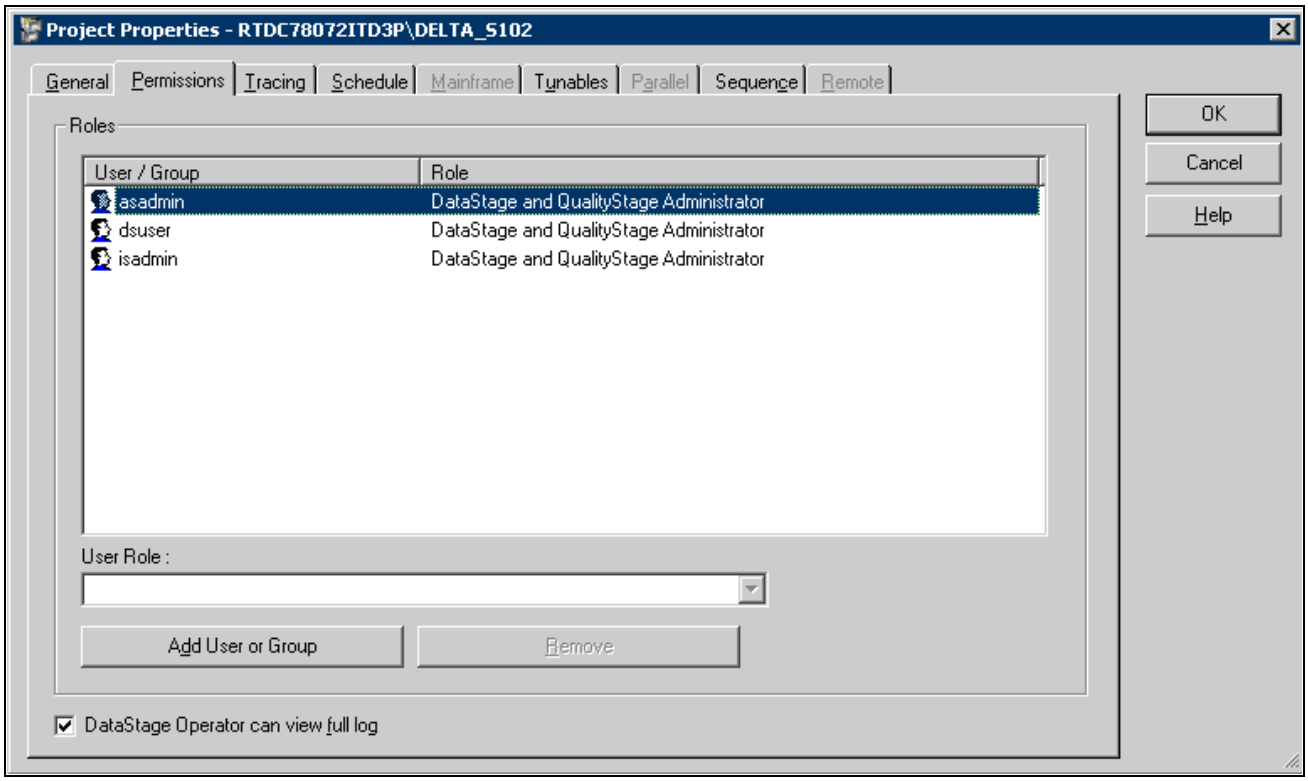
   You should add values to the environment parameters to successfully run an ETL job.



Environment Variables Window - Add Values

Project Properties - Permissions Tab

Access the Project Properties - Permissions tab:



Project Properties - Permissions Tab

Before any user can access WebSphere DataStage they must be defined in the Suite Administrator tool as a DataStage Administrator or a DataStage User. As a DataStage administrator you can define whether a DataStage user can access a project, and if so, what category of access they have.

Use the Permissions tab to add groups and assign users to groups. These groups are in turn allocated the role of DataStage Administrator or DataStage User. Any users belong to an administrator group will be able to administer WebSphere DataStage. You can also grant user group access to a project and assign a role to the group.

When setting up users and groups, these still have to have the correct permissions at the operating system level to access the folders in which the projects reside.

The Permissions page contains the following controls:

- Roles: this window lists all the users and groups who currently have access to this project and lists their roles. Note that this window will always include users who have been defined as DataStage Administrators in the Suite Administrator tool, and you cannot remove such users from the list or alter their user role.
- User Role: this list contains the four categories of WebSphere DataStage user you can assign. Choose one from the list to assign it to the user currently selected in the roles window.
- Add User or Group: click this to open the Add Users/Groups dialog box in order to add a new user or group to the ones listed in the roles window.
- Remove: click this to remove the selected user or group from those listed in the roles window.

- DataStage Operator can view full log: by default this check box is selected, letting a WebSphere DataStage operator view both the error message and the data associated with an entry in a job log file. To hide the data part of the log file entry from operators, deselect this check box. Access to the data is then restricted to users with a developer role or better.

Project Properties - Tracing Tab

Access the Project Properties - Tracing tab:

Project Properties - Tracing Tab

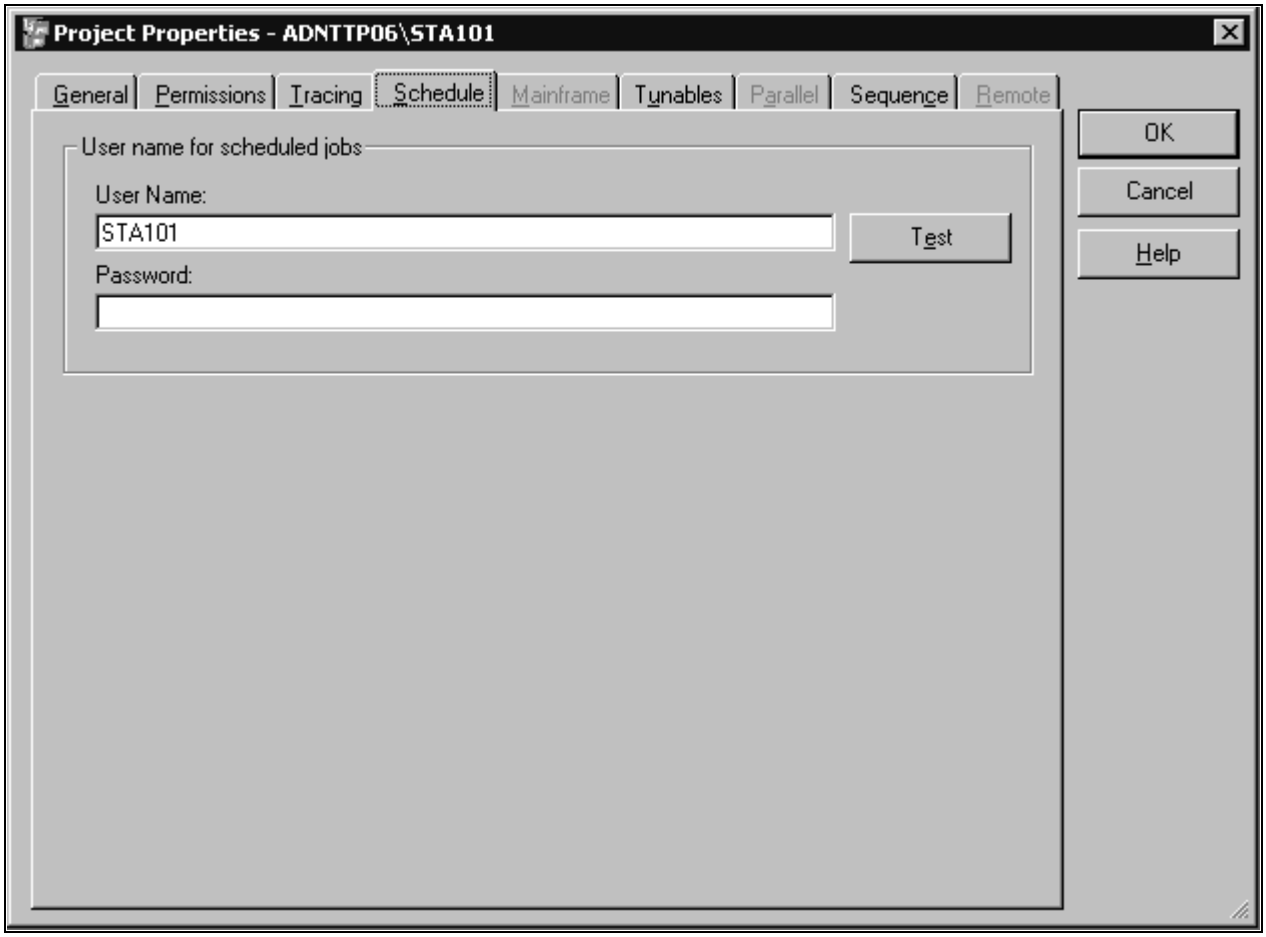
Use the Project Properties - Tracing tab to enable or disable tracing, and view or delete trace files.

Tracing helps you diagnose project problems. Enabling tracing activity on the server helps diagnose project problems. By default, server tracing is disabled.

When you enable tracing, server activity attached to a specific project is written to trace files. Users can use the information saved in trace files to identify the cause of a project problem.

Project Properties - Schedule Tab

Access the Project Properties - Schedule tab:



Project Properties - Schedule Tab

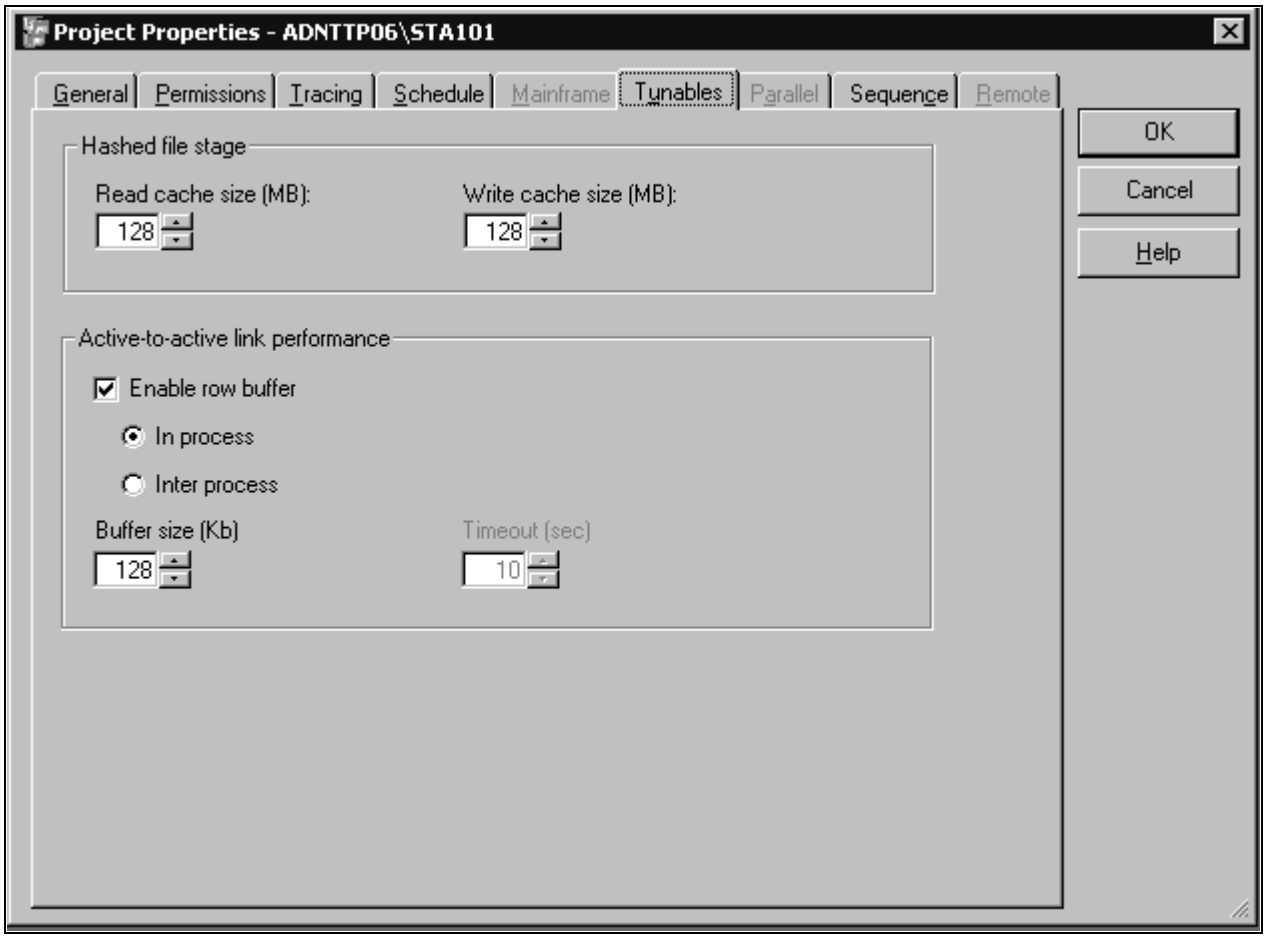
Use the Project Properties - Schedule tab to modify system authority user name for scheduling jobs. DataStage uses the Microsoft Windows Schedule service to schedule jobs. By default, jobs run under the Microsoft Windows system authority user name. However, this user name may not have enough rights, so you may need to change the assigned user name.

To verify that the user name exists, click the Test button. The system schedules and runs a job using the name that you entered.

Note. The Schedule tab is only available on Microsoft Windows.

Project Properties - Tunables Tab

Access the Project Properties - Tunables tab:



Project Properties - Tunables Tab

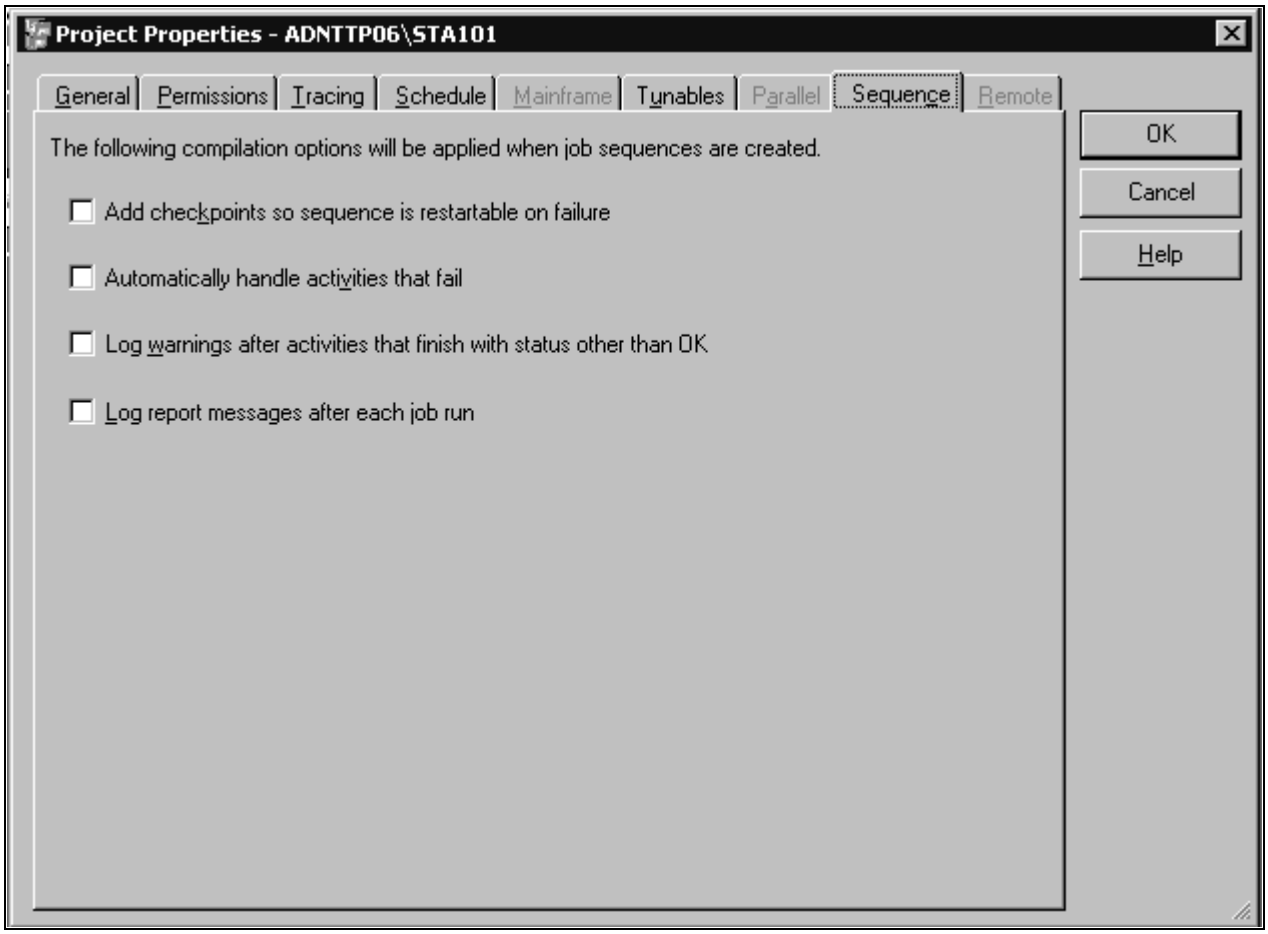
Use the Project Properties - Tunables tab to set up caching details for hashed file stages and row buffering to improve the performance of server jobs.

When data is referenced repeatedly, for instance in a lookup, storing the data in memory rather than on disk can improve performance. To support this performance improvement, when a hash file stage writes records to a hash file, the data can be cached rather than written to the hash file immediately. Similarly, when a hash file stage is reading a hash file, you can preload the file to memory, which makes subsequent access to the data faster. The hash file stage area of the Tunables tab enables you to adjust the sizes of both the read and write cache sizes.

Another way to improve performance is with the use of row buffering. Row buffering enables connected active stages to pass data by using buffers (memory) rather than passing data row by row.

Project Properties - Sequence Tab

Access the Project Properties - Sequence tab:



Project Properties - Sequence Tab

Use the Project Properties - Sequence tab to add checkpoints to a job sequence and enable automatic handling of failures during sequence runs.

You can insert checkpoints in job sequences to enable the sequence to be restarted if one of the jobs in the sequence fails. Checkpoints enable you to see where the problem is, fix it, and then rerun the sequence from the point at which it left off.

Chapter 9

Using DataStage Designer

This chapter provides an overview of DataStage Designer and discusses how to:

- Manage Repository Objects
- Edit Object Properties
- Import and Export Repository Components
- Use Table Definitions
- Build DataStage Jobs
- Use Database and File Stages
- Add and Link Stages
- Compile and Run Jobs
- Edit Job Properties
- Use Expressions
- Create Constraints
- Use Hashed File Stages
- Use Job Sequencers
- Use DataStage BASIC

Note. This chapter does not discuss all the features available for DataStage Designer. For a complete view of DataStage Designer functionality, please see the delivered IBM WebSphere documentation.

DataStage Designer Overview

The DataStage Designer is the primary interface to the metadata repository and provides a graphical user interface that enables you to view, edit, and assemble DataStage objects from the repository needed to create an ETL job.

An ETL job should include source and target stages. Additionally, your server job can include transformation stages for data filtering, data validation, data aggregation, data calculations, data splitting for multiple outputs, and usage of user-defined variables or parameters. These stages allow the job design to be more flexible and reusable.

DataStage Designer enables you to:

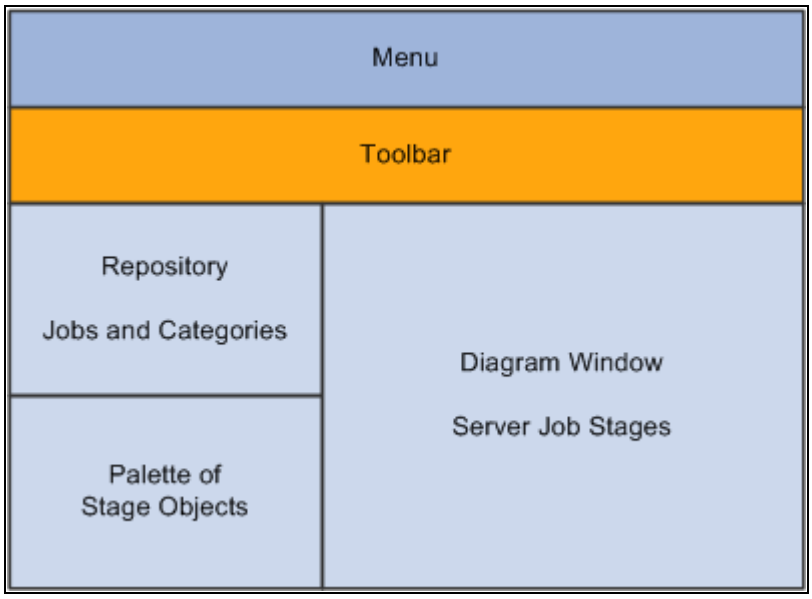
- Create, edit, and view objects in the repository.
- Create, edit, and view data elements, table definitions, transforms, and routines.
- Import and export DataStage components, such as projects, jobs, and job components.
- Analyze the use of particular items in a project.
- Edit and view user-defined object properties.
- Create jobs, job sequences, containers, and job templates.
- Create and use parameters within jobs.
- Insert and link stages into jobs.
- Set stage and job properties.
- Load and save table definitions.
- Save, compile, and run jobs.

DataStage Designer Window

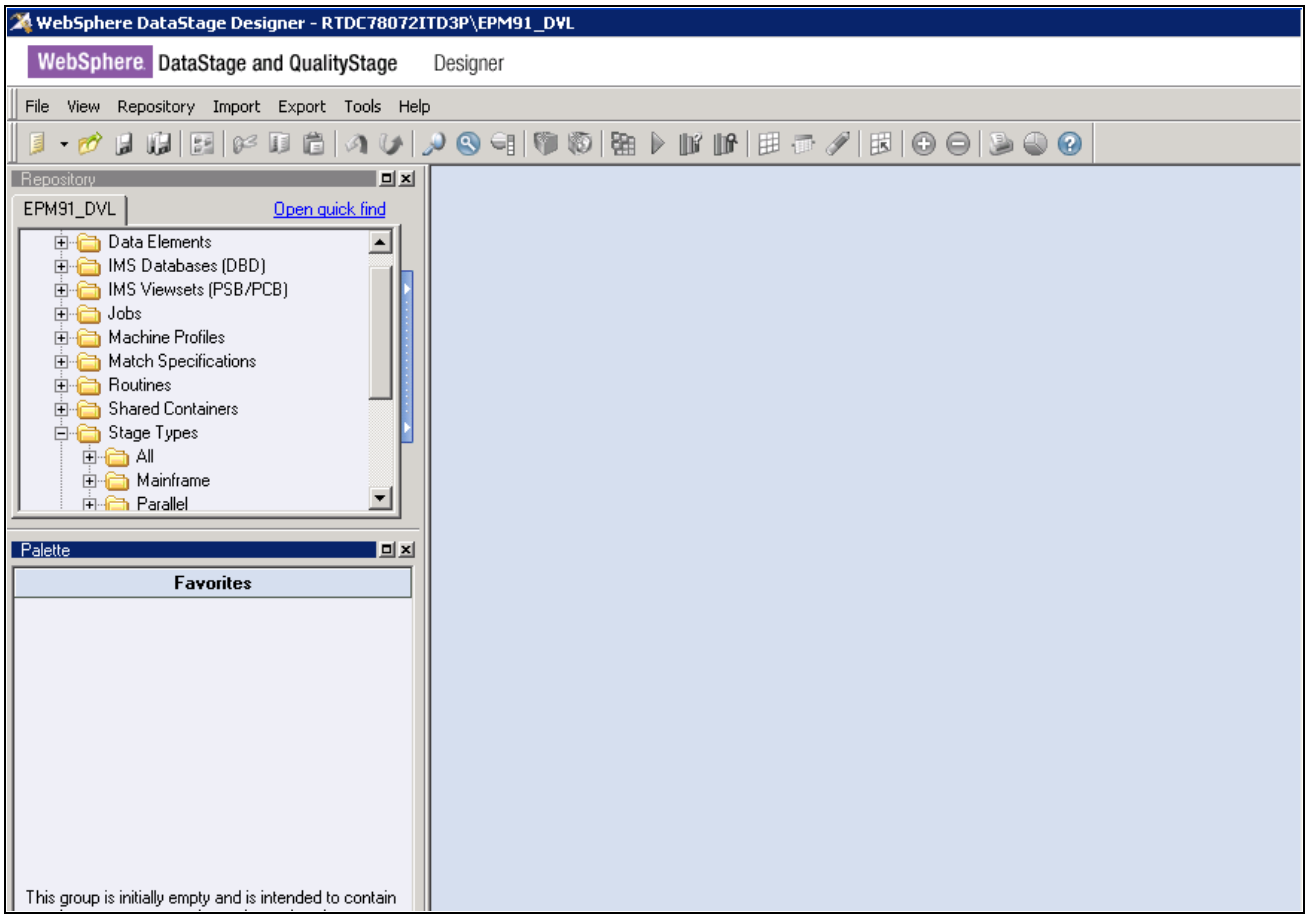
The DataStage Designer window, which is the graphical user interface used to view, configure, and assemble DataStage objects, contains the following components:

- *Repository Window:* Displays project objects organized into categories. By default, the Repository window is located in the upper left corner of the Designer window. The project tree displays in this pane and contains the repository objects belonging to a project.
- *Tool Palette:* Contains objects that you add to your job design, such as stage types, file types, database types, and processor objects. You can drag these objects from the Palette into the Diagram window. By default, this window is displayed in the lower left corner, of the Designer window. This window appears to be empty until you open or create a job.
- *Diagram Window:* Serves as the canvas for your job design. You drag, drop, and link stages and processor objects to create jobs, sequencers, and templates.
- *Property Browser:* Displays the properties of the currently selected stage of the job that is open in the Diagram window. By default, this window is hidden. To open it, select View, Property Browser from the menu bar, and then click a stage to see its properties.

The following diagrams show the layout of the DataStage Designer window components:



DataStage Designer Window - Layout View



DataStage Designer Window

The display area is in the right pane of the DataStage Designer window and displays the contents of a chosen object in the project tree.

By Default, the Designer window contains the Repository window, Tool Palette, and Diagram window. You can optionally view the Property Browser by selecting View, Property Browser from the menu bar .

The display of Designer windows and toolbars can be shown or hidden by selecting the appropriate option from the View menu. You can dock, undock, or rearrange the Designer windows.

Designer Menus

Most Designer menu items are also available in the toolbars. The following are some additional options that are available through the menus:

Designer Menu Item	***Description***
View, Customize Palette	Customize your palette.
View, Property Browser	Enables you to view and edit properties of a DataStage object.

Designer Menu Item	Description
Import	Enables you to import ETL projects, jobs, or other components that you export from another system, as well as DataStage components, such as table definitions, from text files or XML documents.
Export	Enables you to export DataStage objects in the form of text files with the file extension *.dsx*.
Tools, Run Multiple Job Compile	Enables you to compile all your jobs at the same time.
Tools, Run Director	Invoke the Director module, and log you into your project automatically.

Designer Toolbar

The Designer toolbar displays the following buttons:

DataStage Designer Toolbar

This table describes the Designer toolbar buttons:

Designer Toolbar Button	Description
New	Open the New window where you can open a new DataStage object.
New (arrow down button)	Display options associated with the New command on the toolbar.
Open	Display the Open window that enables you to open an existing or recently opened repository object.
Save	Save the current job or container.
Save All	Save all open jobs or containers.
Job Properties	Open the Job Properties window for the current job open in the Diagram window.
Cut	Cut a specific object or text and temporarily stores it.
Copy	Copy a specific object or text and temporarily stores it.
Paste	Paste the temporarily stored object or text.
Undo	Undo the last task performed.

Designer Toolbar Button	***Description***
Redo	Redo the last task performed.
Quick Find	Search for DataStage objects using the quick find feature.
Advanced Find	Search for DataStage objects using the advanced find feature.
Data Flow Analysis	Use this function to display the data lineage for a column definition to see where in the job design that the column definition is used, display the source of the data for selected column or columns, display the target for the data for selected column or columns.
Construct Local Container	Create a local job container.
Construct Shared Container	Create a shared container reusable by other jobs.
Compile	Compile the current job.
Run	Run the current job.
Grid Lines	Show or hide a grid in the Diagram window.
Link Markers	Show or hide markers on the links.
Toggle Annotations	Show or hide annotations in the diagram window. You enter annotations by dragging the Annotation object from the Palette.
Stage Validation errors	See visual cues for parallel jobs or parallel-shared containers. The visual cues display compilation errors for every stage on the canvas, without you having to actually compile the job. The option is enabled by default
Snap to Grid	When the grid is shown and Snap to Grid is enabled, align objects that you drag with the grid.
Zoom In	Magnify the diagram display.
Zoom Out	Shrink the diagram display.
Print	Print the current diagram window.
Generate Report	Generate an HTML report of a server, parallel, or mainframe job or shared container. You can view this report in a standard Internet browser.
Help on View	View context-sensitive help.

Debug Toolbar

The Debug toolbar provides basic functions for testing and troubleshooting your jobs.

The Debug toolbar can be accessed by selecting View, Debug and displays the following buttons:

Debug Toolbar

This table describes the Debug toolbar buttons:

Debug Toolbar Button	*Description*
Set target debug job	Enables you to select the job you want to debug.
Start/Continue Debugging	Start or stop running in debug mode.
Next Link	Run the job until you come to the next link.
Next Row	Run until you get to the next row.
Stop Job	Stop the job run.
Set debug Job Parameters	Set job parameters.
Edit Breakpoints	Change breakpoints (pauses that you have inserted into the run).
Toggle Breakpoints	Enable or disable breakpoints.
Clear All Breakpoints	Clear breakpoints.
View Job Log in Director	Open the job log in the Director module.
Show/Hide Debug Window	Display or hide the debug window.

All of the Debug toolbar options are also available from the Debug menu.

Managing Repository Objects

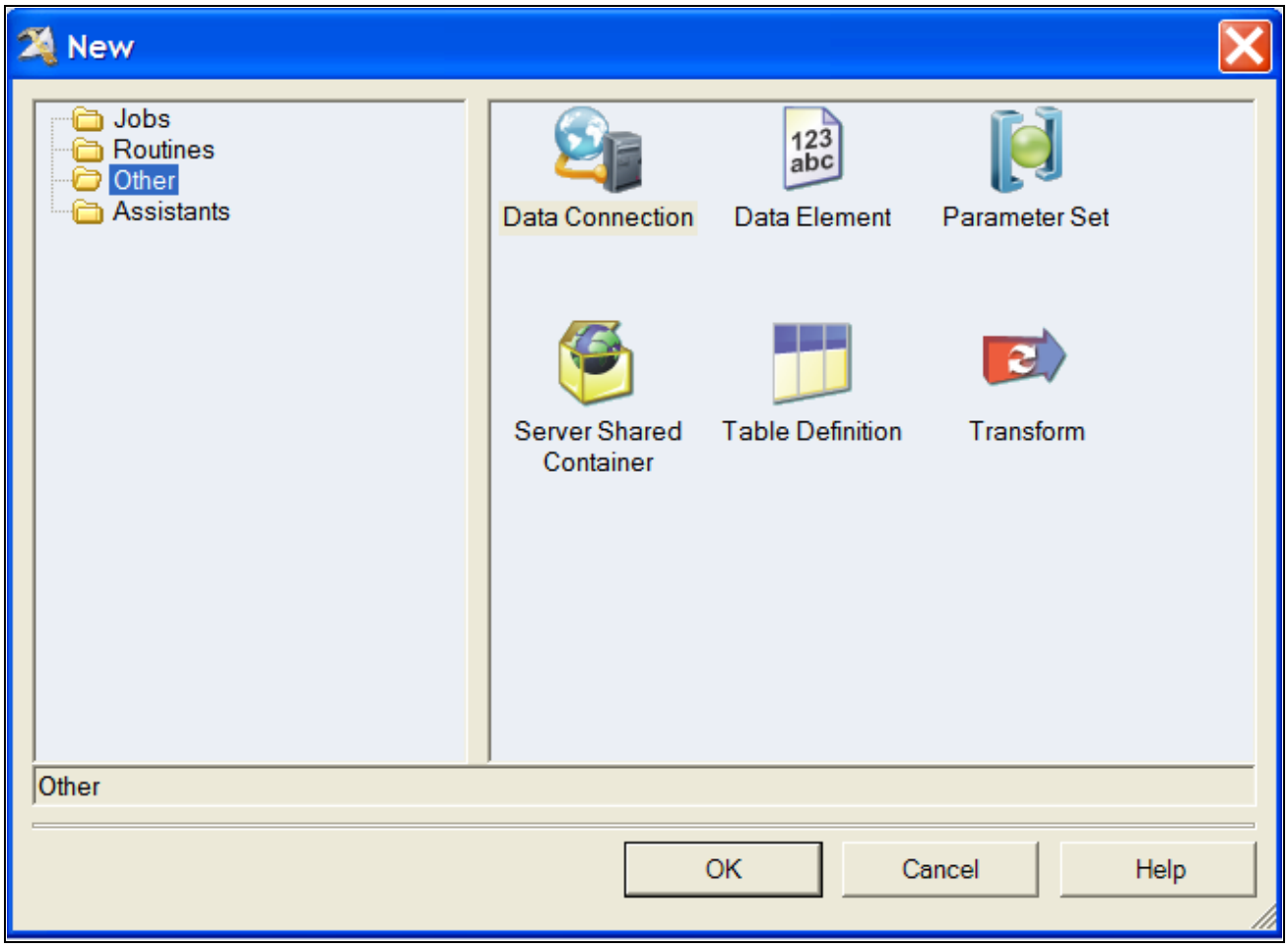
You can use DataStage Designer to view job categories, which serve to organize repository objects.

You can view the following repository objects within a job category:

- Data Elements.

- Jobs.
- Routines.
- Shared Containers.
- Stage Types.
- Table Definitions.
- Transforms.

You can also create new repository objects:



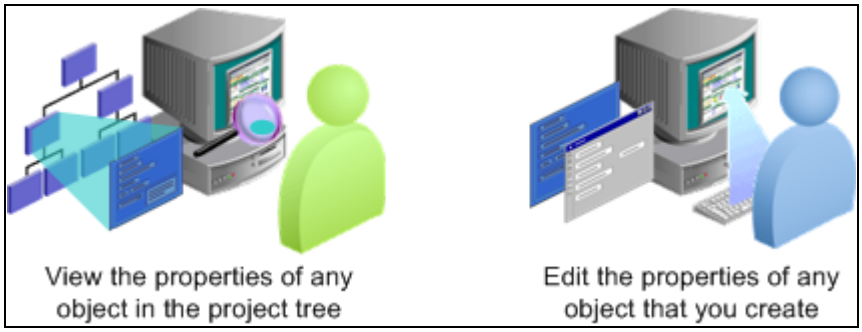
Create new Data Element

You can also copy, rename, edit, delete, or move an item using the File menu commands or the item level shortcut menu.

Editing Object Properties

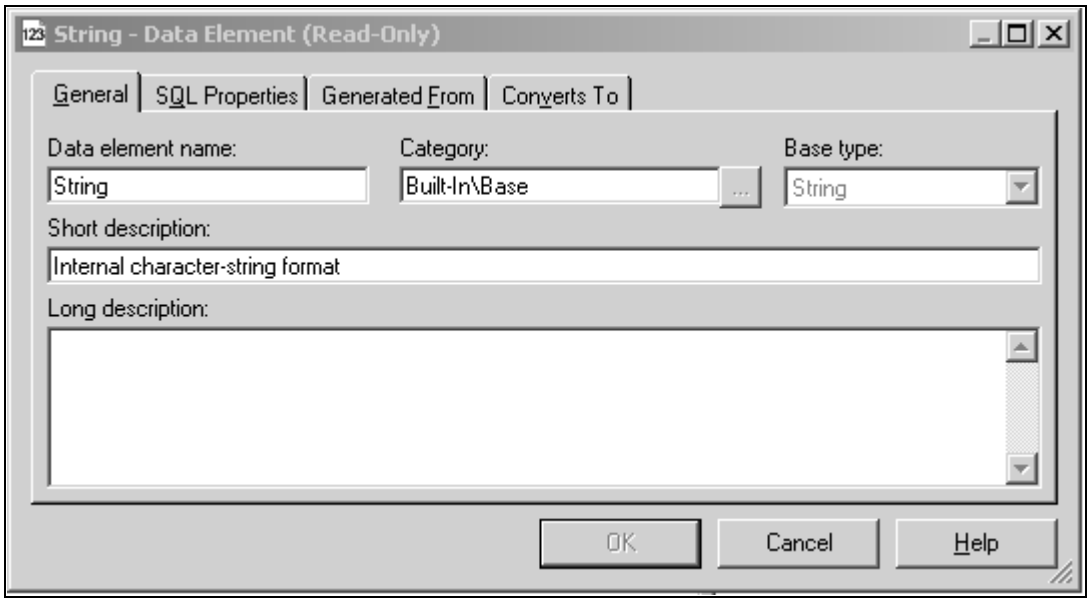
Object properties consist of descriptive information and other types of information, depending on the object type.

Using DataStage Designer you can:



DataStage Designer - Object Properties

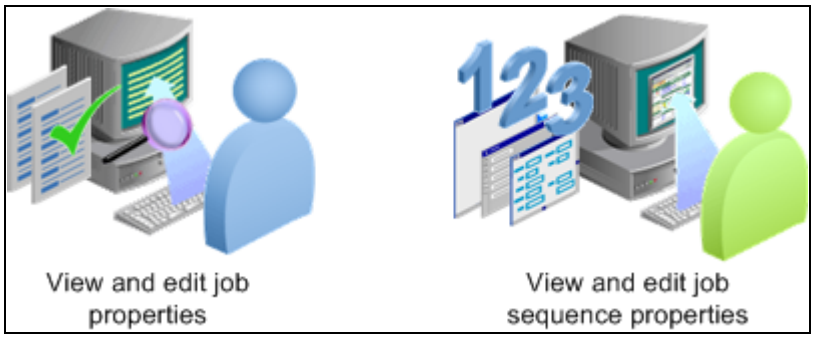
The following is an example of an object property for the String data element:



Object Property - String Data Element

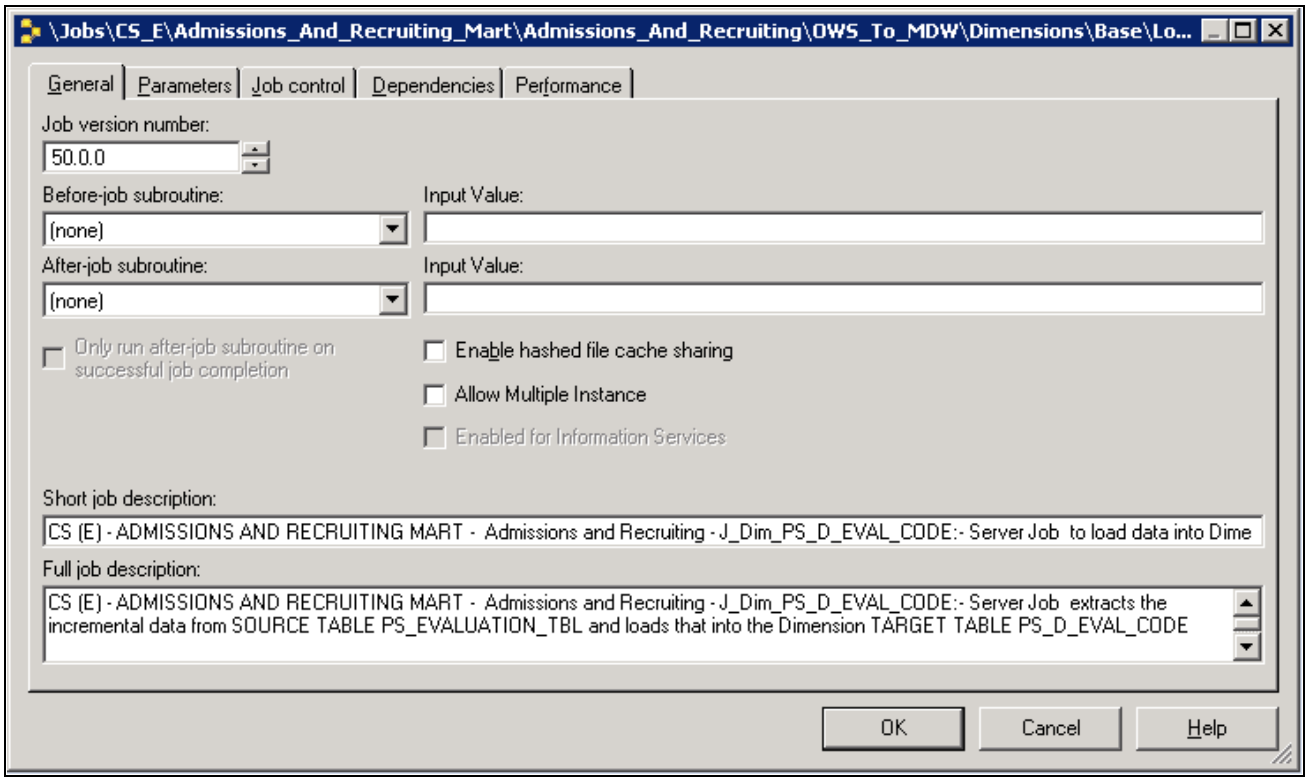
Editing Job and Job Sequence Properties

DataStage Designer enables you to:



DataStage Designer - Job/Job Sequence Properties

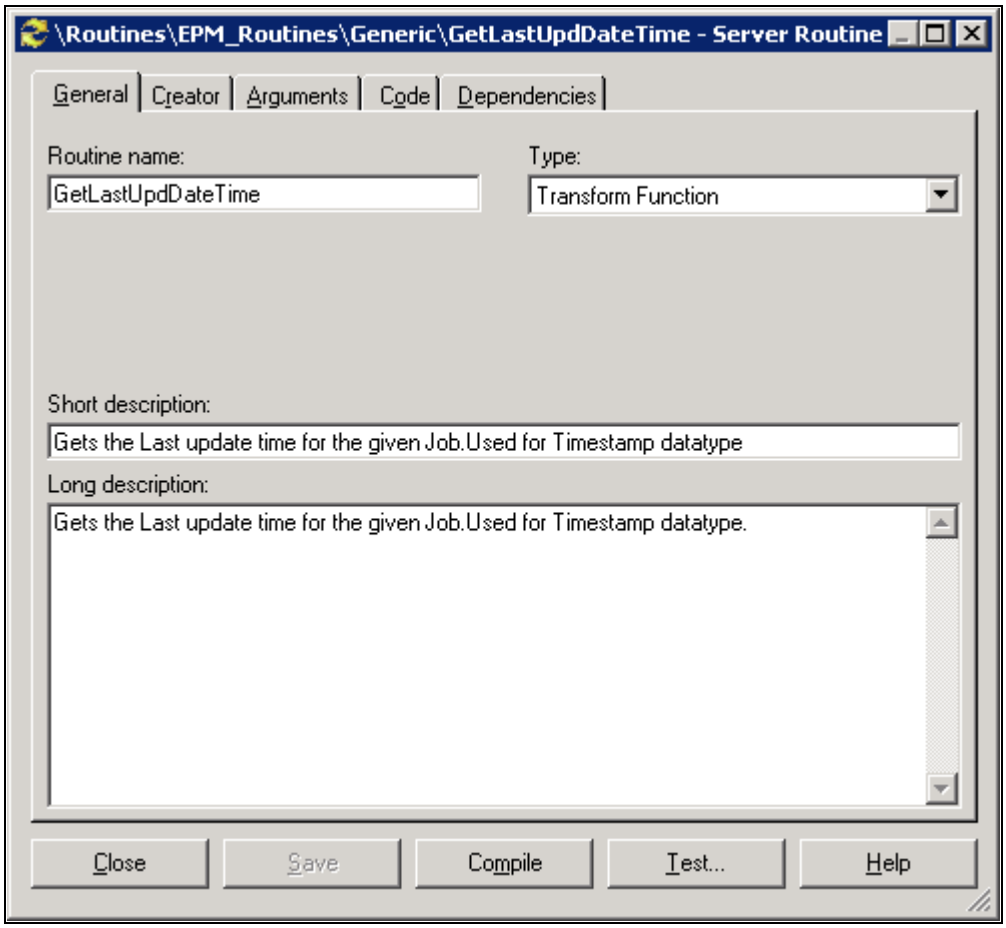
The following is an example of a server job property:



Server Job Properties

Editing Server Routines

You can create, edit, or view server routines using the Routine window. Argument names in built-in routines cannot be changed.



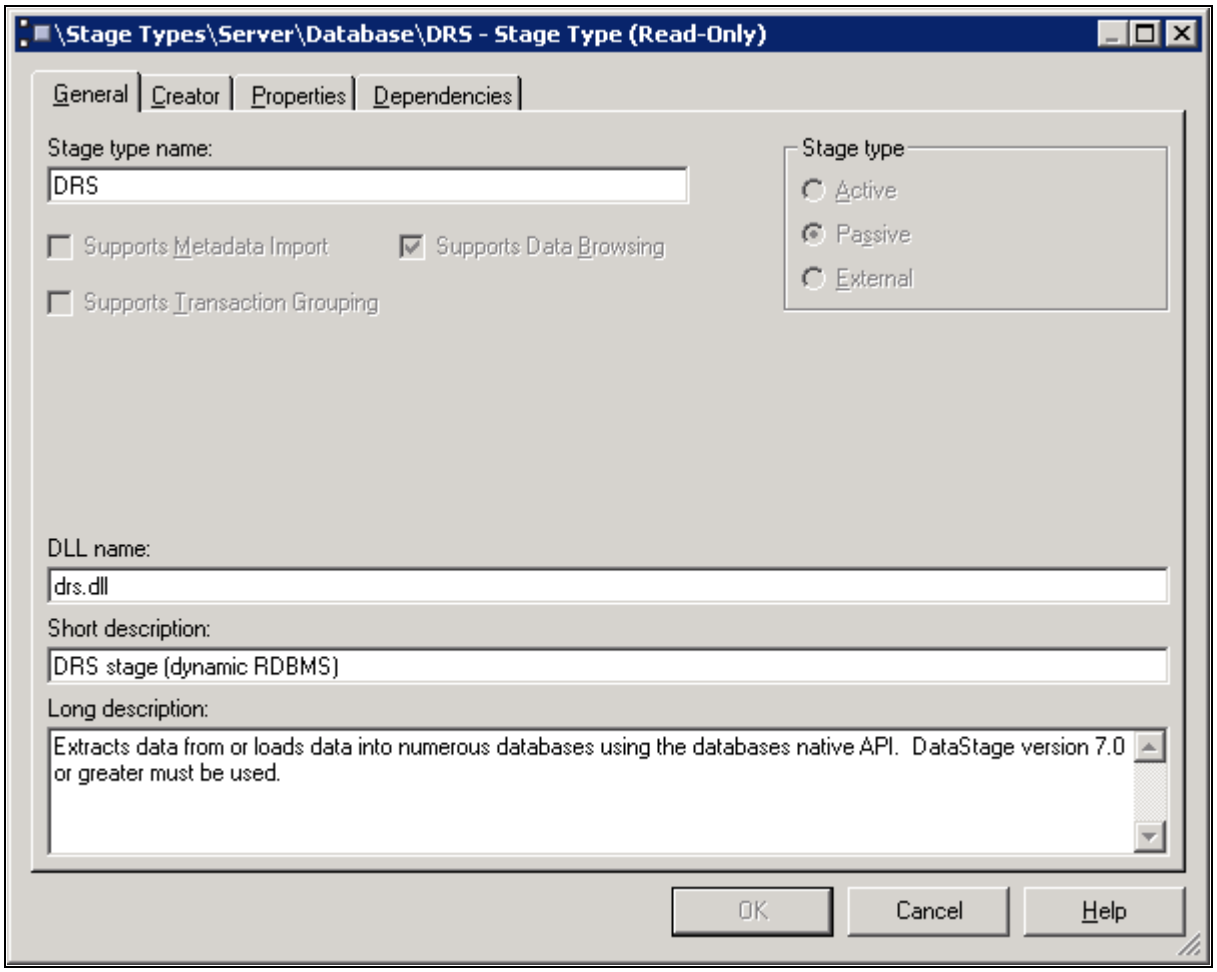
Server Routine Properties

The following components are classified as routines:

- Transform functions.
- Before/after subroutines.
- Custom UniVerse functions.
- ActiveX (OLE) functions.

Editing the Stage Type

The Stage Type category in the project tree contains all the stage types that you can use in your jobs. Properties of WebSphere DataStage's pre-built stages are read-only.



DRS Stage Type Properties

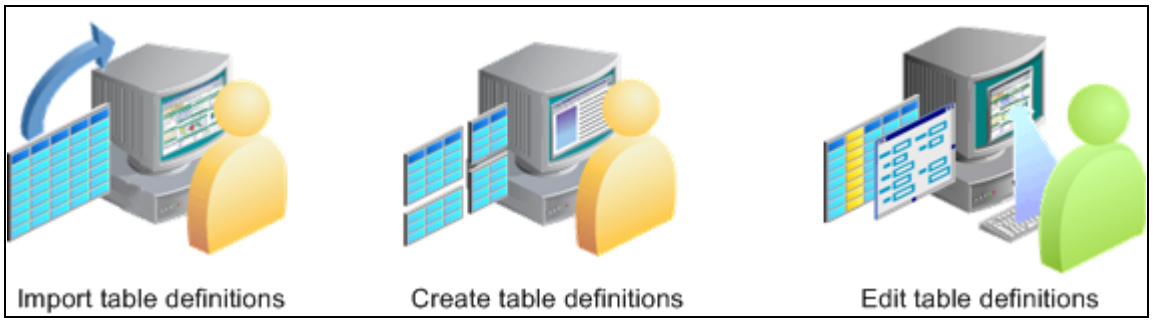
You can create or edit object properties for the following stage types:

- Custom Plug-in Stages.
- Parallel Job Custom Stages.

DataStage Designer enables you to create and register plug-in stages to perform specific tasks that the built-in stages do not support. You need to register custom plug-in stages before you can use them. In addition, DataStage Designer enables you to create custom parallel stage types.

Specifying Table Definitions

DataStage Designer enables you to:



DataStage Designer - Table Definitions

Table definitions:

- Specify the data structure used by each stage in a DataStage job.
- Are stored in the repository and are shared by all jobs in a project.
- Are required for each data source and data target.
- Can be imported, manually created, or edited.

Importing and Exporting Repository Components

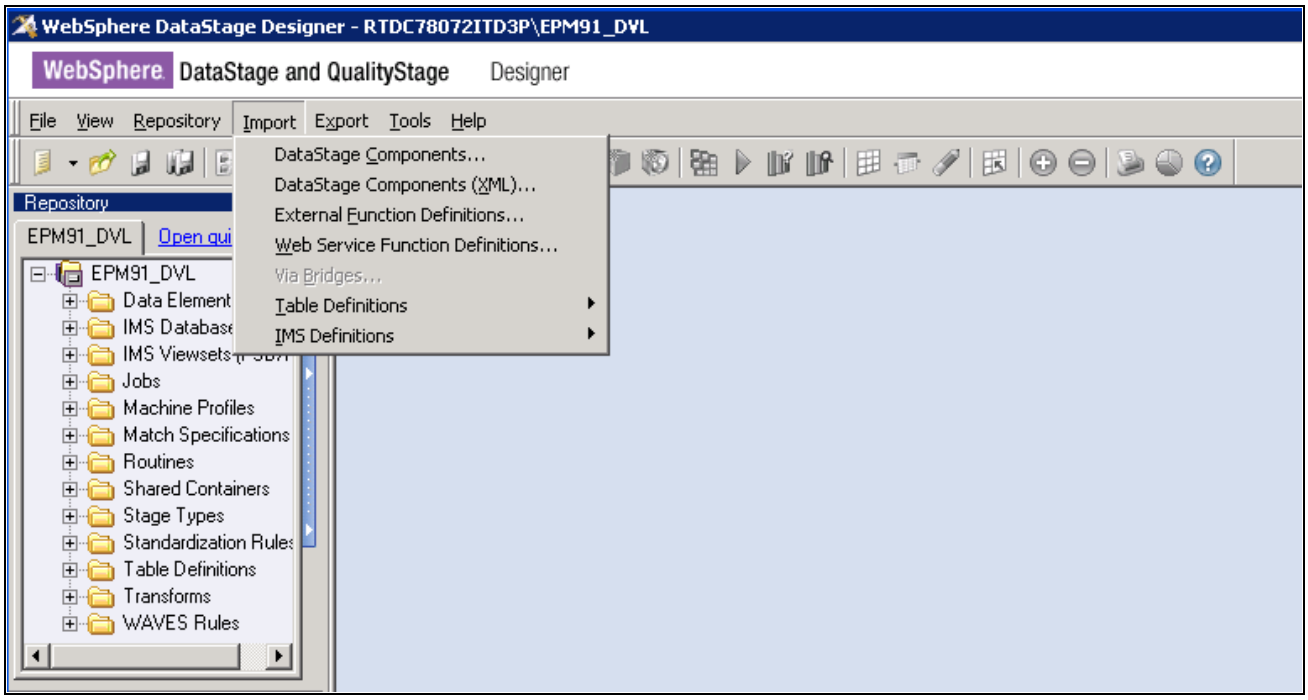
Using the DataStage Designer import and export facilities enable you to move jobs or other components between projects. You can also move projects, jobs, or components from one system to another. In addition, you can import components from text files or XML documents, and you can export to XML documents. XML documents can be used as a convenient way to view descriptions of repository objects using a web browser.

Importing

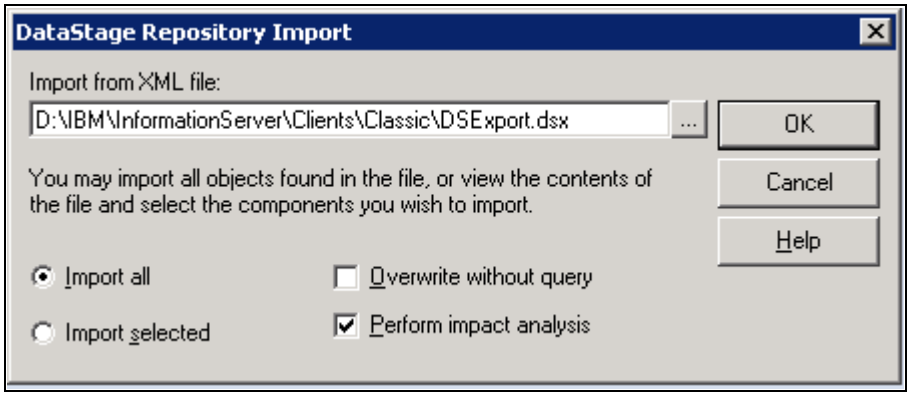
The DataStage Designer import facility enables you to import:

- ETL projects, jobs, or other components that you export from another system.
- DataStage components, such as table definitions, from text files or XML documents.

You can use the Import facility to import table definitions from a variety of file types, including sequential files, ODBC, and XML.



Import Menu Item



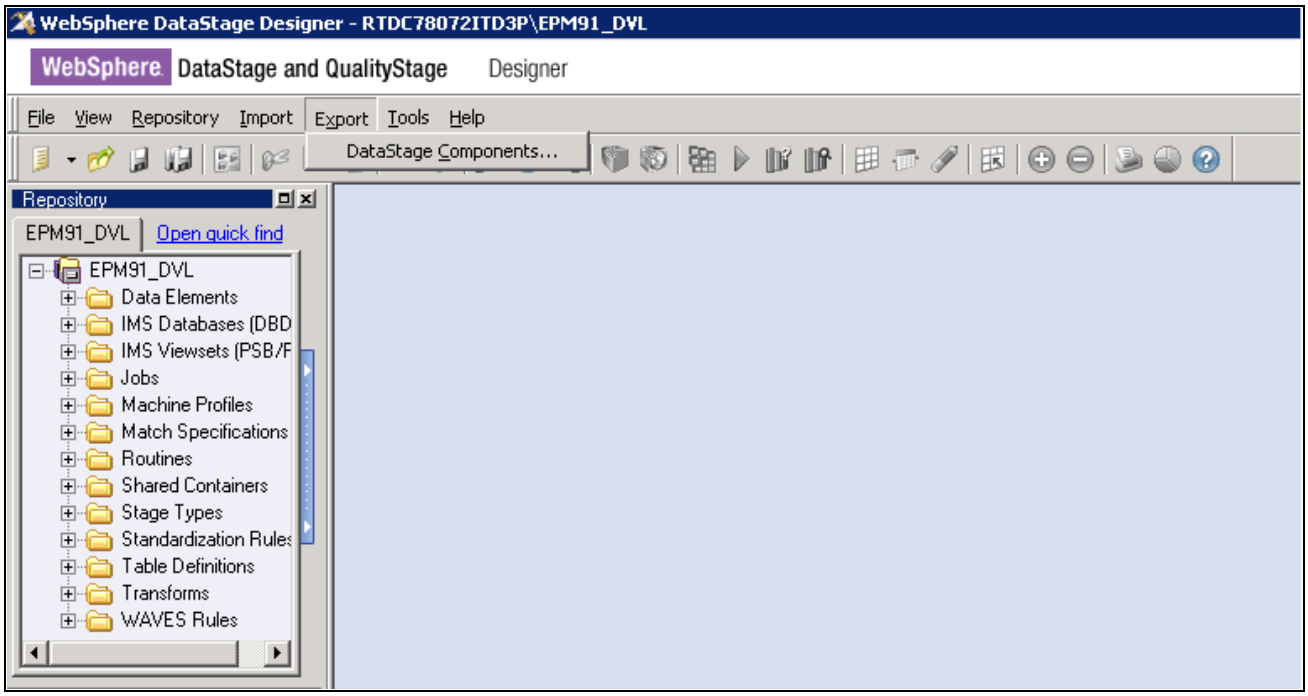
Import DSX Component

Exporting

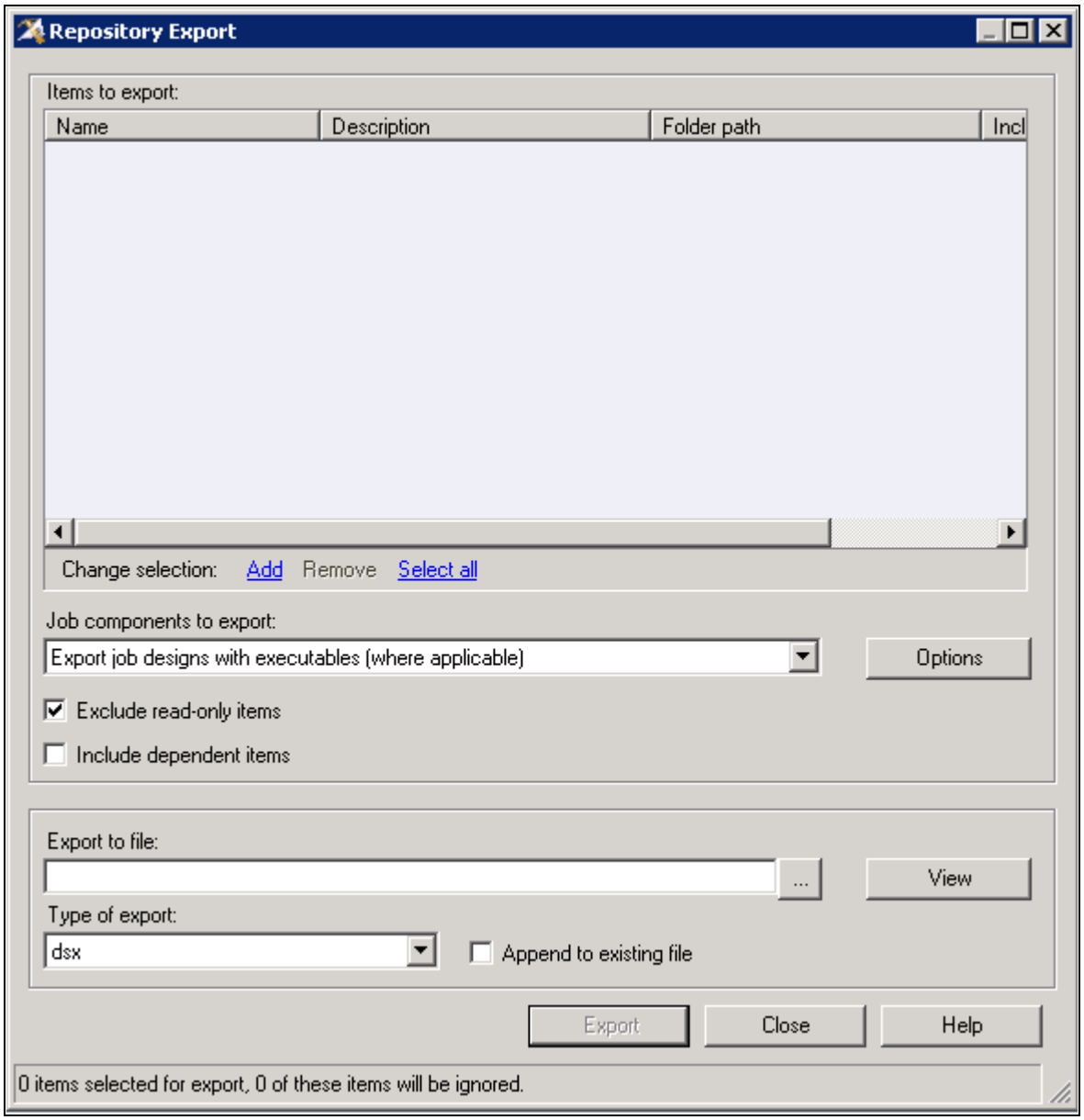
The DataStage Designer export facility enables you to export:

- ETL projects, jobs, or other components.
- Jobs or other components to XML documents.
- Job executables.
- Package server jobs using the Packager Wizard.

When you export projects or components, by default they are stored in text files with the file extension .dsx. You can also export to XML files by selecting the appropriate check box in the Export window. You also have the option to append the exported items to an existing file.



Export Menu Item



Export - Repository Export window

Using Table Definitions

Table definitions are:

- DataStage components that specify the metadata used at each stage of a job.
- Stored in the Repository.
- Shared by all the jobs in a project.

You need a table definition for each data source stage or data target stage you use in your job. You can import, create, or edit a table definition using DataStage Designer.

Creating Table Definitions

To create a new Table Definition, select New Table Definition from the Table Definition menu. The Table Definition window appears:

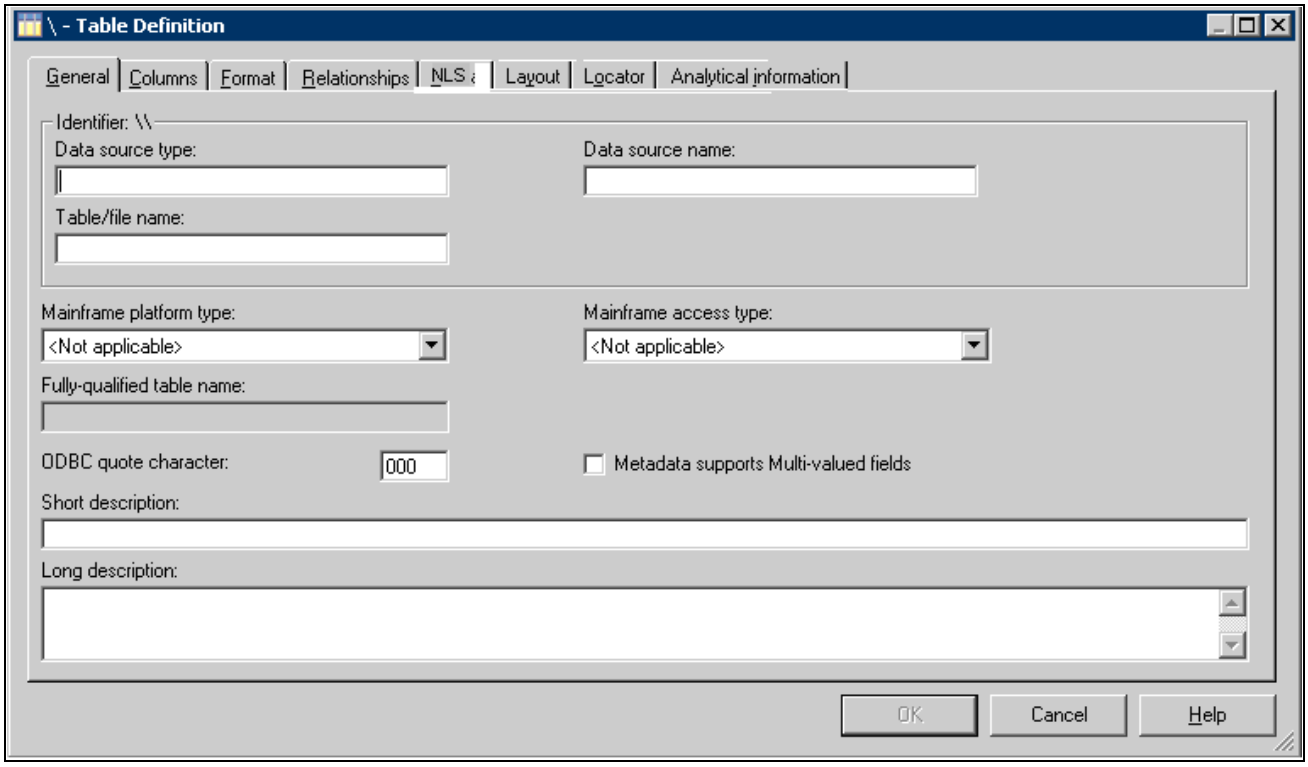


Table Definition Window

The Table Definition window has these tabs:

- General

  The General tab contains the data source type, data source name, table or file name, and other general information about the table definition.

- Columns

  The Columns tab contains a grid displaying the column definitions for each field in the table definition.

- Format

  The Format tab contains file format parameters for sequential files used in DataStage jobs

- Relationships

  The Relationships tab displays the details of any relationship this table definition has with other tables, and allows you to define new relationships.

- NLS

  If NLS is enabled, the NLS tab is enabled and contains the name of the map to use for the table definition.

- Layout

  The Layout tab displays the schema format of the column definitions in a table.

- Locator

  Using the Locator tab you can view and edit the data resource locator associated with the table definition. The data resource locator is a property of the table definition that describes the real world object from which the table definition was imported. The labels and contents of the fields in this window depend on the type of data source or target from which the locator originates.

- Analytical Information

  The Analytical Information tab displays information about the table definition generated by Information Analyzer.

- Parallel

  The Parallel tab displays detailed format information for the defined metadata for parallel jobs.

Importing Table Definitions

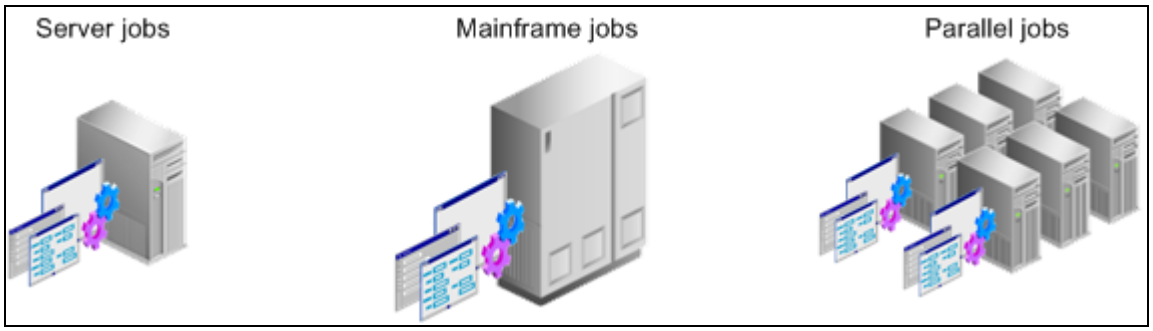
You can directly import a table definition from a source or target database. You can import table definitions from ODBC data sources, plug-in stages, UniVerse tables, hash files, UniData files, or sequential files.

In the DataStage Designer Repository window, right-click on Table Definitions. Select Import.

You can select the type of table definition data source from the available options.

Building DataStage Jobs

DataStage provides these types of jobs:

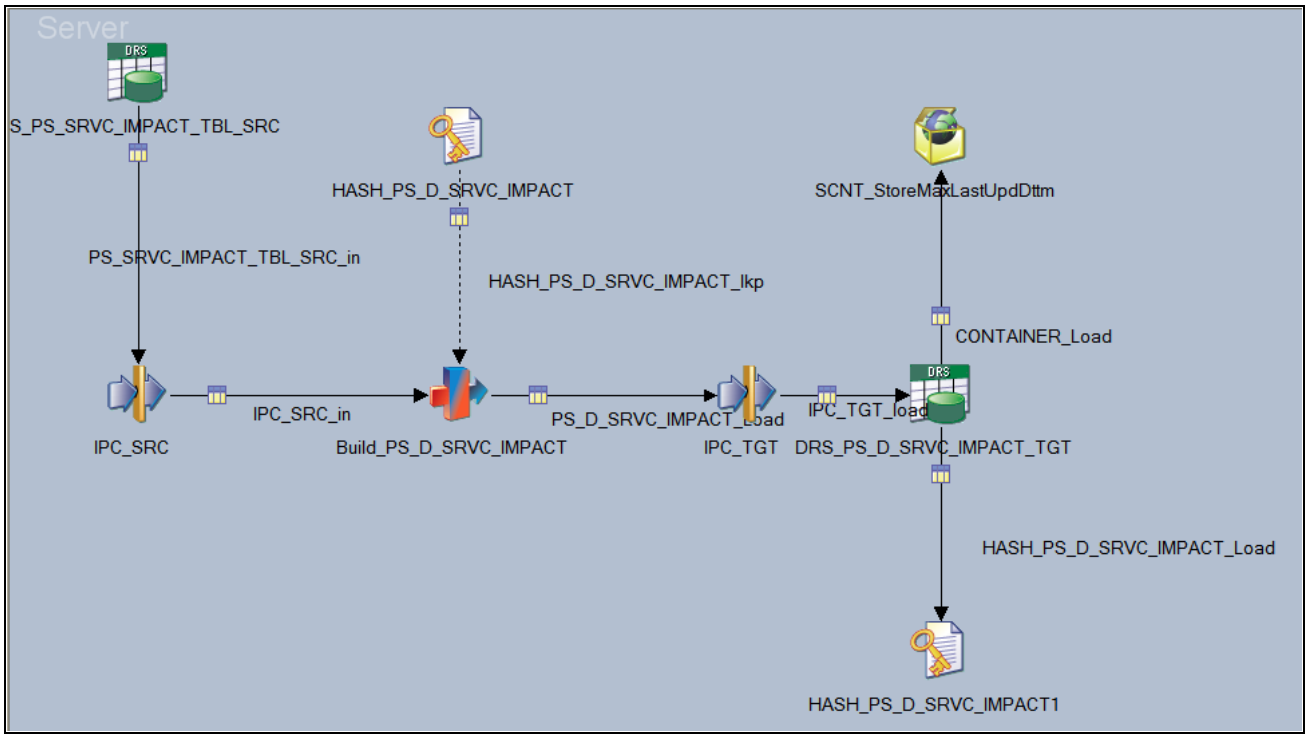


DataStage Jobs

- Server Jobs: Run on the DataStage Server.
- Mainframe Jobs: Available only if you have installed Enterprise MVS Edition and uploaded it to a mainframe, where they are compiled and run.

- Parallel Jobs: Available only if you have installed the Enterprise Edition and run on DataStage servers that are SMP, MPP, or cluster systems.

The following is an example of one of the delivered Campus Solutions Warehouse server jobs:



Sample Delivered Job - J_Dim_PS_D_SRVC_IMPACT

Perform the following steps to build a job:

1. Define optional project-level environment variables in DataStage Administrator.
2. Define optional environment parameters.
3. Import or create table definitions, if they are not already available.
4. Add stages and links to the job to indicate data flow.
5. Edit source and target stages to designate data sources, table definitions, file names, and so on.
6. Edit transformer and processing stages to perform various functions, include filters, create lookups, and use expressions.
7. Save, compile, troubleshoot, and run the job.

Using Database and File Stages

Database stages represent data sources or data targets.

DataStage provides three types of stages:

- Server Job Database Stages
- Server Job File Stages
- Dynamic Relational Stages
- Processing Stages

Each stage has a set of predefined and editable properties.

Server Job Database Stages

The following are some of the delivered server job database stages:

- ODBC
- UniVerse
- UniData
- Oracle
- Sybase

Server Job File Stages

The delivered server job file stages are:

- Sequential file
- Hashed file
- Complex flat file
- Folder

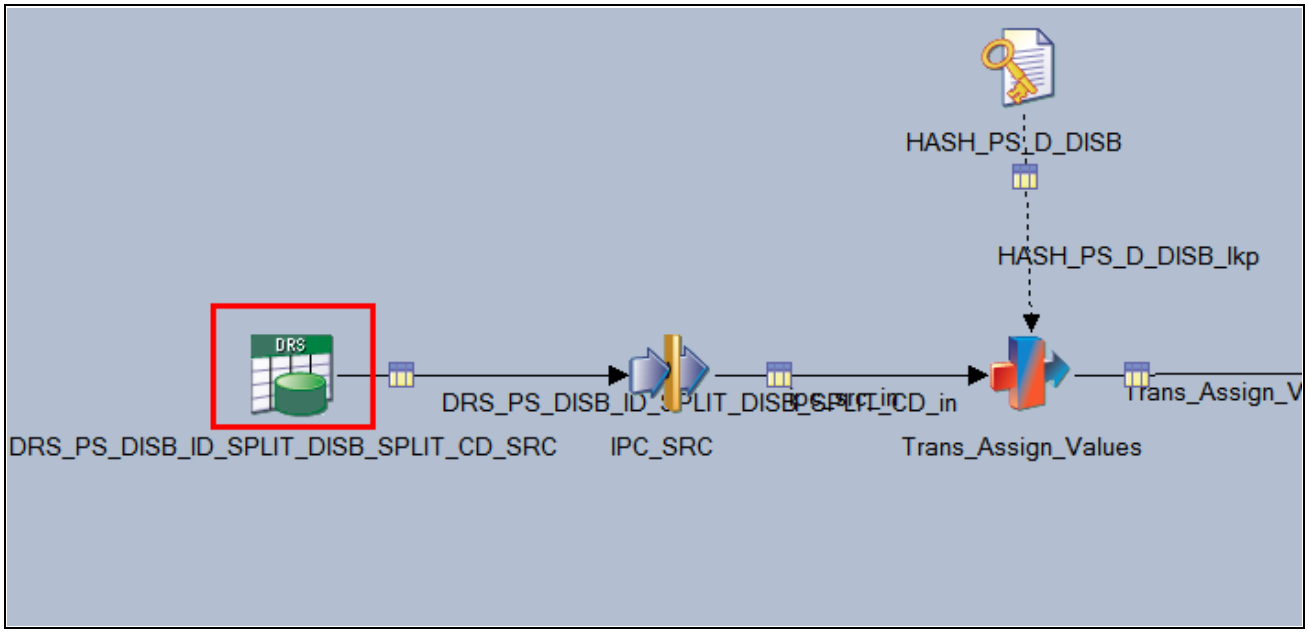
Dynamic Relational Stages

Dynamic Relational Stages (DRS):

- Read data from any DataStage stage.
- Read data from any supported relational database.
- Write to any DataStage stage.
- Write to any supported relational database.

PeopleSoft-delivered ETL jobs use the DRS stage for all database sources or targets. This is represented in the Database group as "Dynamic RDBMS." When you create jobs, it is advisable to use the DRS stage rather than a specific type such as DB2 because a DRS will dynamically handle all of PeopleSoft supported database platforms.

The following example shows a DRS database stage in a delivered Campus Solutions Warehouse job:



DRS Stage

A DRS database stage supports the following relational databases:

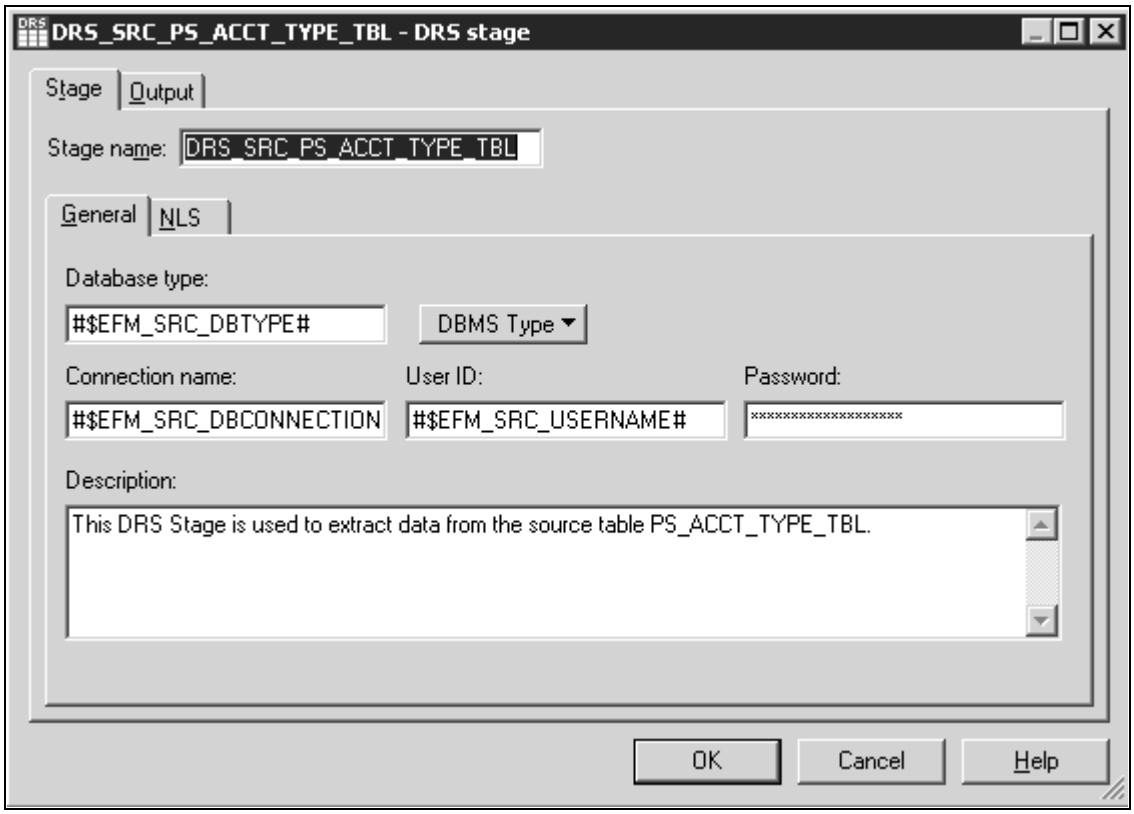
- DB2/UDB
- Informix
- Microsoft SQL Server
- Oracle
- Sybase

A DRS database stage also supports any generic ODBC interface.

Editing the DRS Stage

You edit the DRS properties using the DRS stage window.

1. Double-click the DRS stage to open the DRS stage window.
2. The DRS stage window contains two main tabs: the Stage tab and the Output tab:

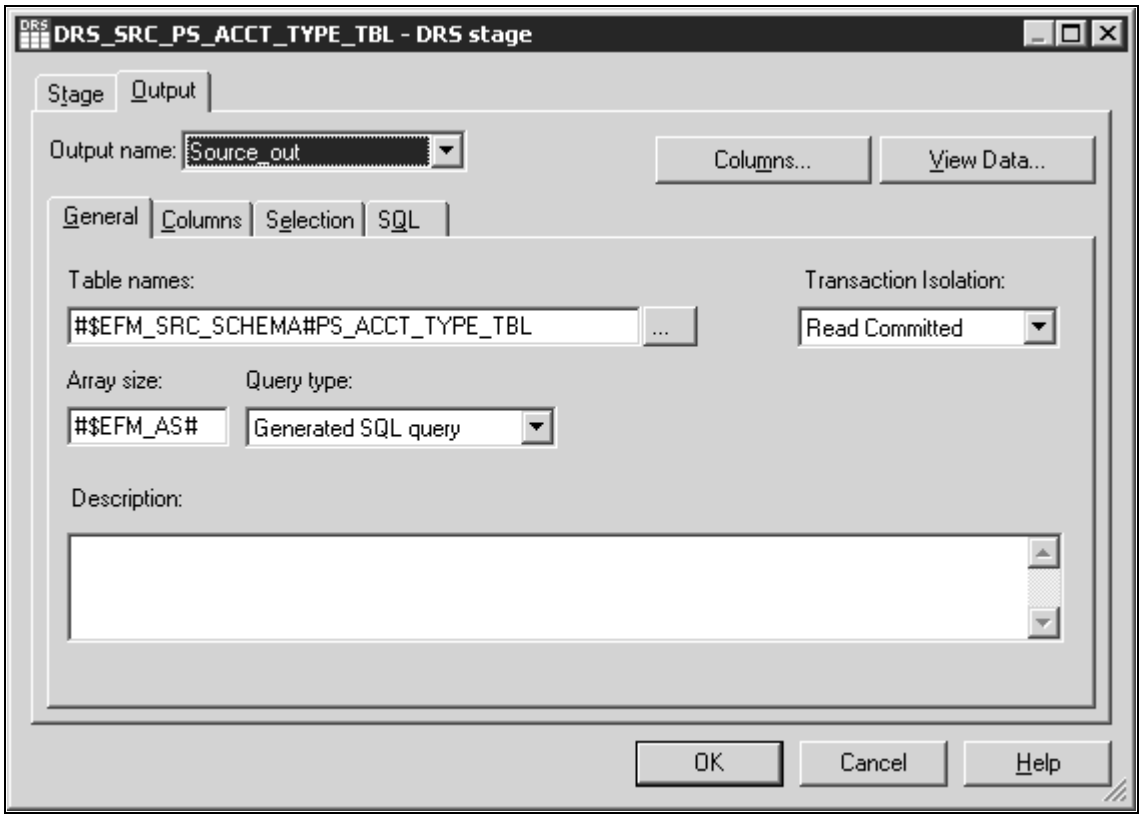


DRS Stage Window

The Stage tab contains two tabs: the General tab and the NLS tab. In the General tab, you define the source database type, database or connection name, user ID, and password used in that connection. The previous example uses environment variables to define the values of these fields. If environment variables or job parameters were not used in the DRS stage, you define the actual values in these fields.

Entering Information in the Output Window

The Output tab contains General, Columns, Selection, and SQL tabs:

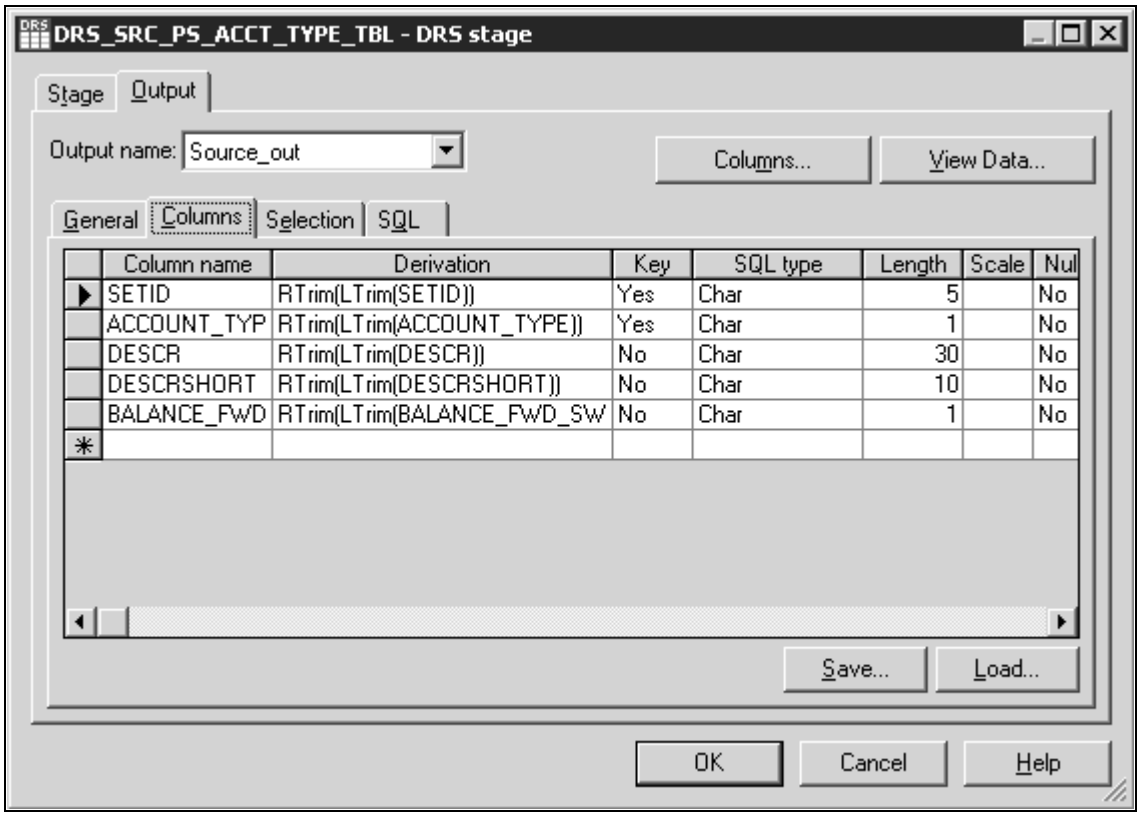


DRS Stage Output Window

In this example, the table name listed is the source of the data that this stage uses.

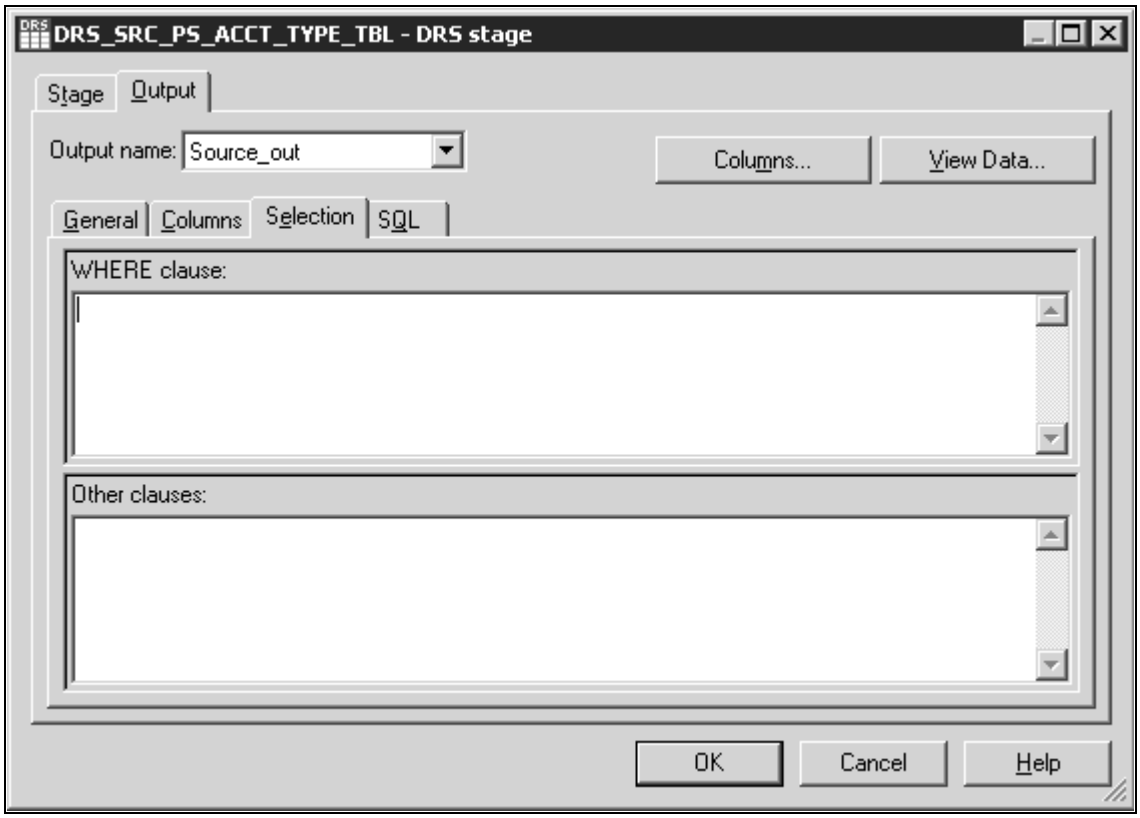
The Columns window shown below enables you to select which columns of data you want to pass through to the next stage. When you click the Load button, the system queries the source table and populates the grid with all the column names and properties. You can then delete rows that are not needed.

The following example shows the Columns window:



DRS Stage Output Window - Columns Tab

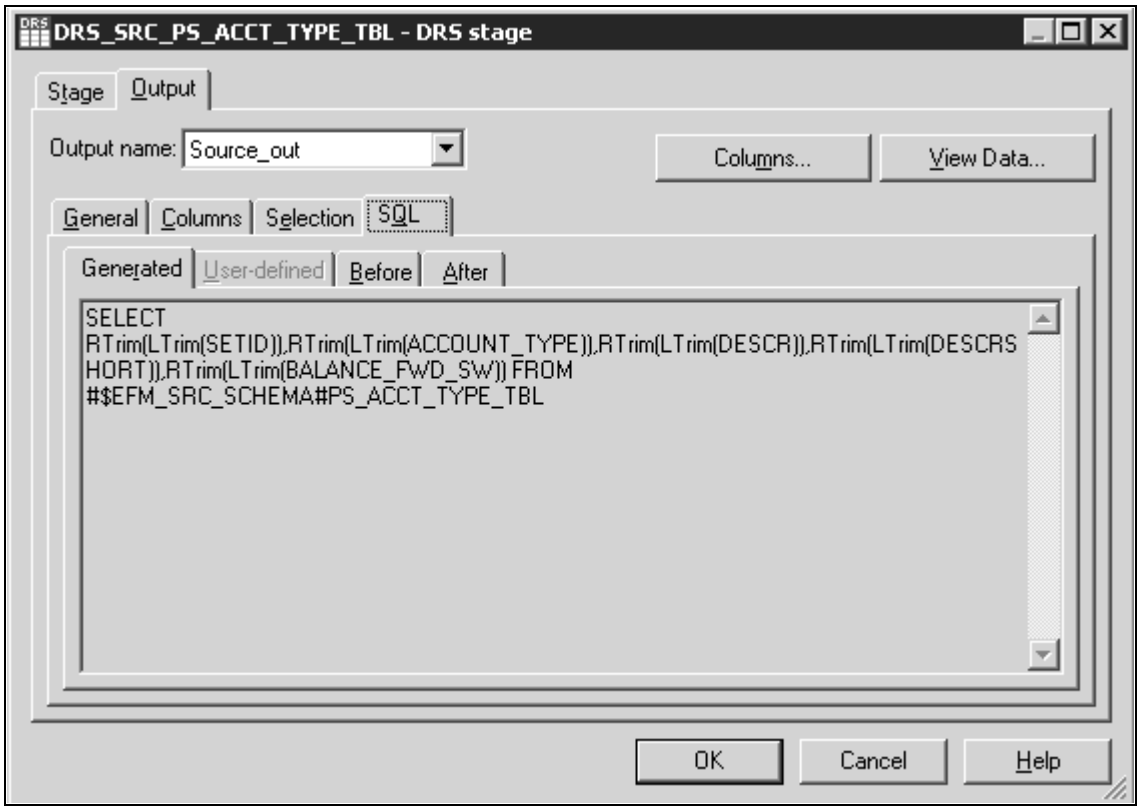
The Selection window enables you to enter a Structured Query Language (SQL) WHERE clause that specifies conditions when fetching data from tables.



DRS Stage Output Window - Selection Tab

Entering a WHERE clause in the Selection window is optional.

The following shows the SQL tab of a DRS stage:



DRS Stage Output Window - SQL Tab

The SQL tab contains the SQL statement used for the current stage.

Window Element	***Usage***
Generated	Shows the SQL SELECT statement that is automatically generated by this stage. It is read-only.
Before	Enter optional SQL statements executed before the stage processes job data rows. This does not appear in every plug-in.
After	Enter optional SQL statements executed after the stage processes job data rows This does not appear in every plug-in.

Note. You can define SQL in a DRS Stage.

Processing Stages

DataStage Processing Stages:

- Reads the data from the source.
- Processes, transforms, or converts the data read from the source.

- Writes the processed data to the target.

Processing Stage Types

This table describes the different types of Processing Stages:

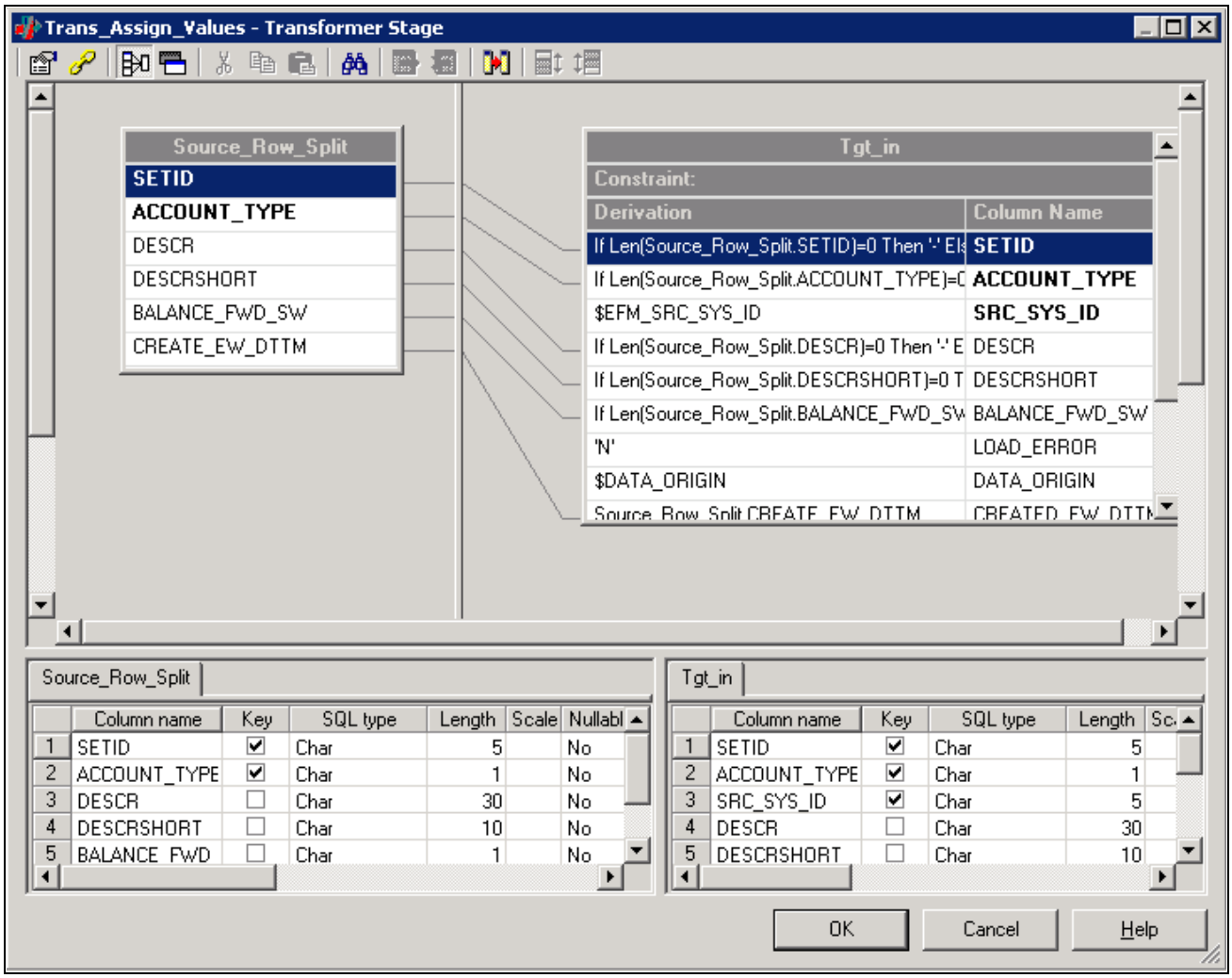
Processing Stage	*Description*
Transformer	Transformer stages perform transformations and conversions on extracted data.
Aggregator	Aggregator stages group data from a single input link and perform aggregation functions such as COUNT, SUM, AVERAGE, FIRST, LAST, MIN, and MAX.
FTP	FTP Stages transfer files to other machines.
Link Collector	Link Collectors collect partitioned data and pieces them together.
Interprocess	An InterProcess (IPC) stage is a passive stage which provides a communication channel between WebSphere DataStage processes running simultaneously in the same job. It allows you to design jobs that run on SMP systems with great performance benefits.
Pivot	Pivot, an active stage, maps sets of columns in an input table to a single column in an output table.
Sort	Sort Stages allow you to perform Sort operations.

Transformer Stages

Transformer stages enable you to:

- Add, delete, or move columns.
- Apply expressions to data.
- Use lookups to validate data.
- Filter data using constraints.
- Edit column metadata and derivations.
- Define local stage variables, and before-stage and after-stage subroutines.
- Specify the order in which the links are processed.
- Pass data on to either another transformer stage, or to a target stage.

The following is an example of a delivered Transformer Stage (Trans_Assign_Values Stage):



Trans_Assign_Values Transformer Stage

Creating Transformer Stages

You create a transformer stage by opening the Processing group in the palette, selecting the Transformer stage, and clicking in the Diagram window. After creating links to connect the transformer to a minimum of two other stages (the input and output stages), double-click the Transformer icon to open the Transformer window.

In the example above, two boxes are shown in the upper area of the window representing two *links*. Transformer stages can have any number of links with a minimum of two. Hence, there could be any number of boxes in the upper area of the window. Labeling your links appropriately makes it easier for you to work in the Transformer Stage window.

The lines that connect the links define how the data flows between them. When you first create a new transformer, you link it to other stages, and then open it for editing. There will not be any lines connecting the Link boxes. These connections can be created manually by clicking and dragging from a particular column of one link to a column in another link, or by selecting the Column Auto-Match button on the toolbar.

Using the Transformer Stage Toolbar

The following buttons appear on the Transformer Stage toolbar:

Transformer Stage Toolbar

This table describes the buttons provided with the Transformer Stage toolbar

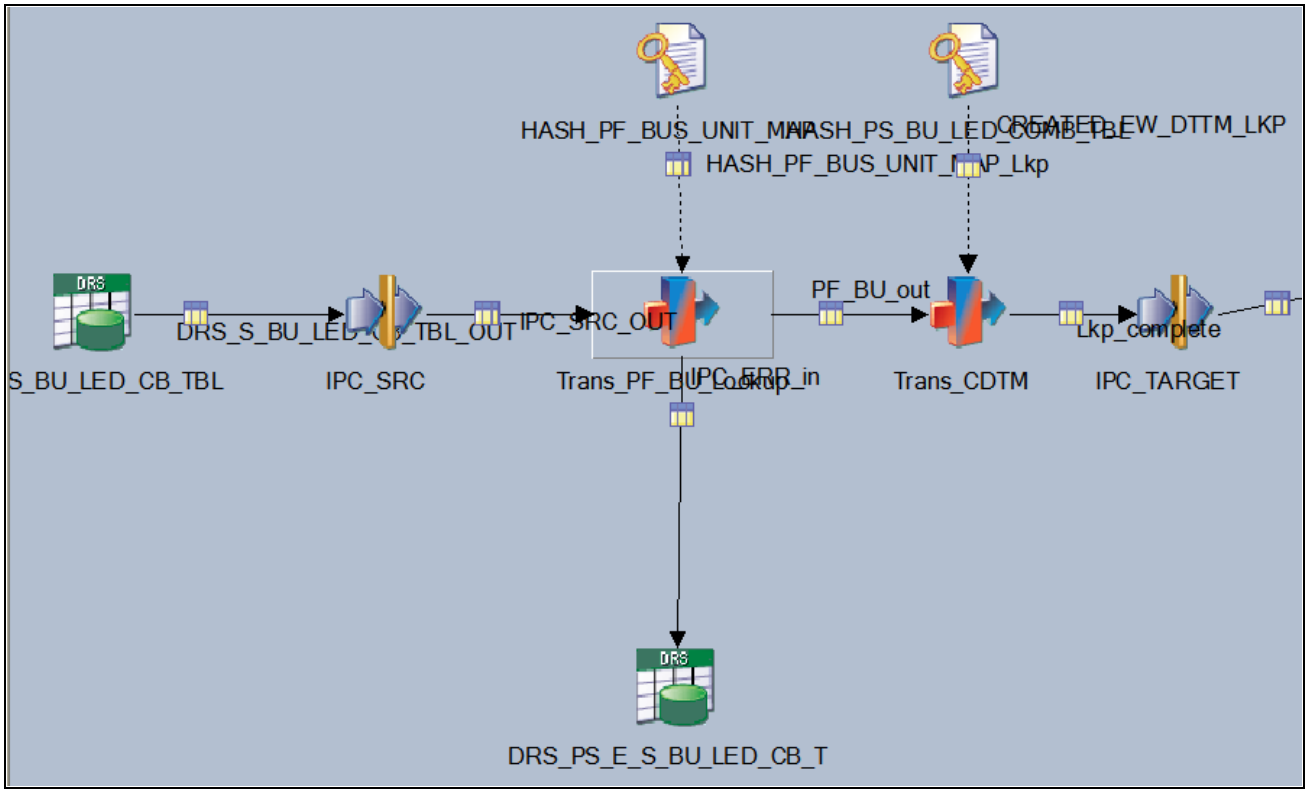
Transformer Toolbar Button	***Usage***
Stage Properties	Define stage inputs and outputs when you link the transformer with other stages. Specify before-stage and after-stage subroutines (optional). Define stage variables. Define order in which input and output links are processed if there is more than one input or output link.
Constraints	Enter a condition that filters incoming data, allowing only the rows that meet the constraint criteria to flow to the next stage.
Show All or Selected Relations	If you have more than two links in the transformer, you can select one link and click this button to hide all connection lines except for those on the selected link. With only two links present, clicking this button hides or displays all connections.
Show/Hide Stage Variables	Show or hide a box that displays local stage variables that can be assigned values in expressions, or be used in expressions.
Cut, Copy, Paste, Find/Replace	These are standard Windows buttons.
Load Column Definition	Load a table definition from the repository, or import a new one from a database.
Save Column Definition	Save a column definition in the repository so that it can be used in other stages and jobs.
Column Auto-Match	Automatically sets columns on an output link to be derived from matching columns on an input link. You can then go back and edit individual output link columns where you want a different derivation.
Input Link Execution Order	Order the reference links. The primary data link is always processed first.
Output Link Execution Order	Order all output links.

Adding and Linking Stages

Stages represent inputs, outputs, and transformations within a job. Links join the stages together and show the flow of data within the job.

You add stages and links to a job by clicking the stage type or link in the palette and then clicking in the diagram window.

The following example shows a job that contains stages and links:



Sample Job with Stages and Links - J_BASE_PS_BU_LED_COMB_TBL

A stage typically has at least one input or one output. However, some stages can have multiple inputs and output to more than one stage.

Different types of job have different stage types. The stages that are available in the DataStage Designer are dependent on the job type that is currently open in the DataStage Designer.

Adding Stages

To add a stage to a job, click a stage type in the palette, and click in the Diagram window.

The stages are located as follows:

- Database stages are located in the Database palette group.
- File stages are located in the File palette group.

- Processing stages are located in the Processing group.

  This group includes the Transformer and Pivot stages used in PeopleSoft-delivered jobs.

If the link is red, then the link is broken. Start and end the drag motion in the center of each stage to ensure that you have linked the stages correctly.

Adding Links

To add a link between stages, you click the Link object in the General palette group, and then click and drag the cursor from one stage to another.

Another option is to right-click on one stage and drag the link to another stage.

By default, new links are named. However, we recommend that you rename all of your links to reflect their purpose and avoid confusion when you are editing transformers and stage properties.

Compiling and Running Jobs

Before running a job you must always:

Compiling a Job

To compile a job, click the Compile button on the DataStage Designer toolbar. After compiling the job, the result appears in the display area. If the result of the compilation is *Job successfully compiled with no errors*, you can schedule or run the job. If an error is displayed, you can click the Show Error button to highlight the stage where the problem occurs. Ensure that you have specified all the input and output column definitions, directory paths, file names, and table names correctly.

Criteria Checked when Compiling Jobs

The link to the source data stage is called the *primary link.* All other input links are called *reference links.*

During compilation, the following criteria in the job design are checked:

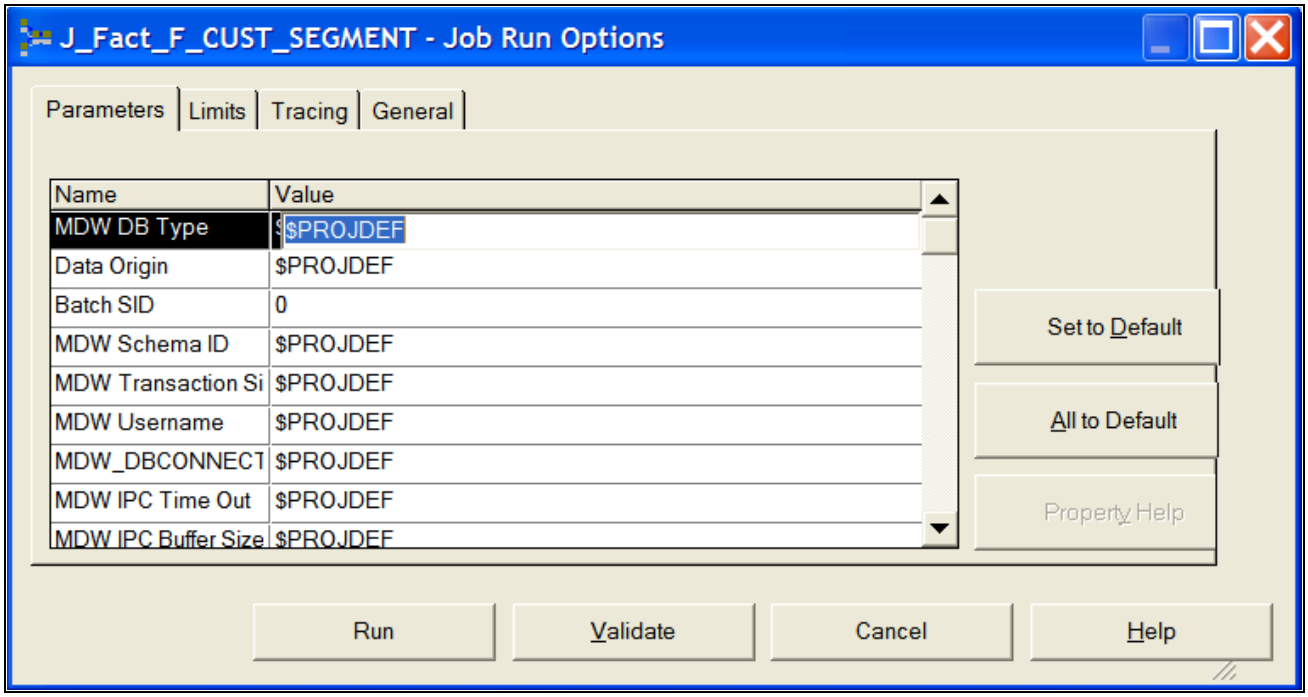
- Primary Input: If you have more than one input link to a Transformer stage, the compiler checks that one is defined as the primary input link.
- Reference Input: If you have reference inputs defined in a Transformer stage, the compiler checks that these are not from sequential files.

- Key Expressions: If you have key fields specified in your column definitions, the compiler checks that there are key expressions joining the data tables.
- Transforms: If you have specified a transform, the compiler checks that this is a suitable transform for the data element.

Specifying Job Run Options

After compiling jobs, they become executable. The executable version of the job is stored in your project along with your job design.

To run a job, click the Run button on the DataStage Designer toolbar. After clicking the Run button the Job Run Options window appears, where you can specify information on running a server job.



Job Run Options Window

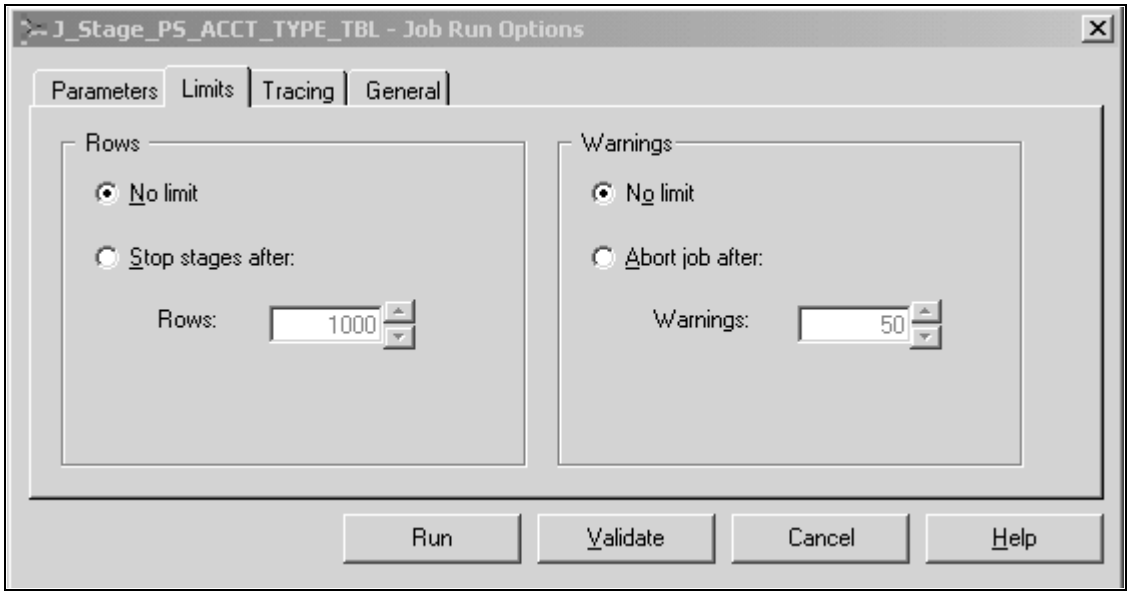
In the Parameters tab, you enter specific parameter values for the job. You specify job parameters in the job properties window. You can create job-specific parameters or use an environment variable defined in DataStage Administrator. When running jobs, the parameters required to run the job are displayed in the Parameters tab of the Job Run Options window. If you specified default values in your job properties, these are displayed in the Parameters tab.

When setting values for environment variables, you can specify either *$PROJDEF*,, *$ENV*, or *$UNSET* special values:

- When you use $ENV, DataStage uses the current setting for the environment variable.
- When you use *$PROJDEF*, the current setting for the environment variable is retrieved and set in the job environment. This allows the environment variable value to be used anywhere in the job. If the value of that environment variable is subsequently changed in DataStage Administrator, the job picks up the new value without the need for recompiling.

- When you use $UNSET, DataStage explicitly unsets the environment variable.

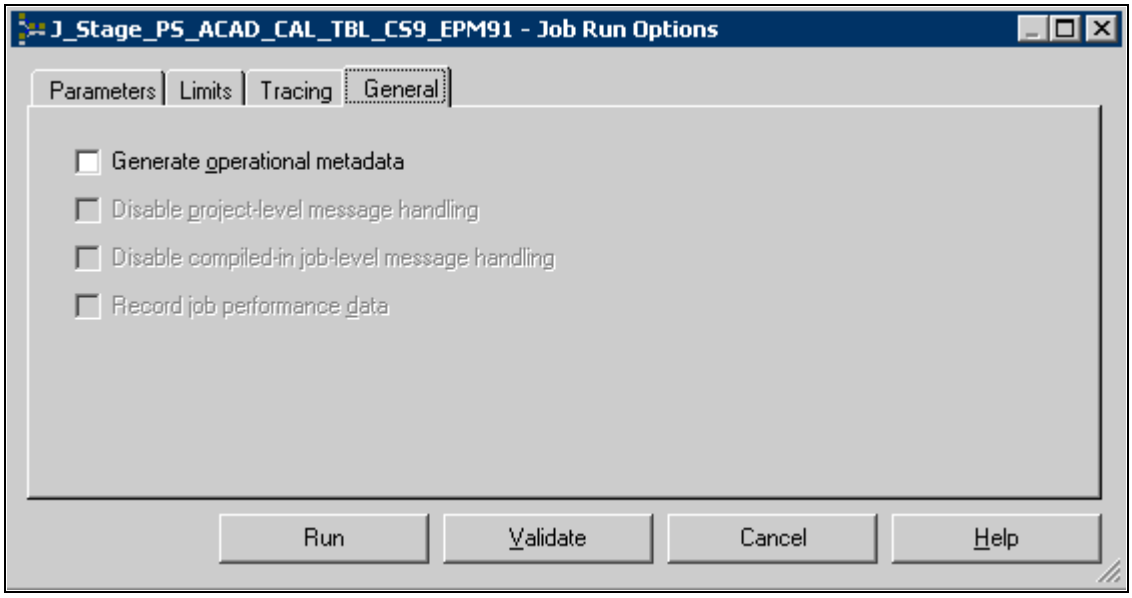
In the Limits tab, you specify any run time limits.



Job Run Options Limits Tab

You can specify whether stages in the job should be limited in how many rows they process and whether runtime error warnings should be ignored.

You specify whether the job should generate operational metadata in the General tab.

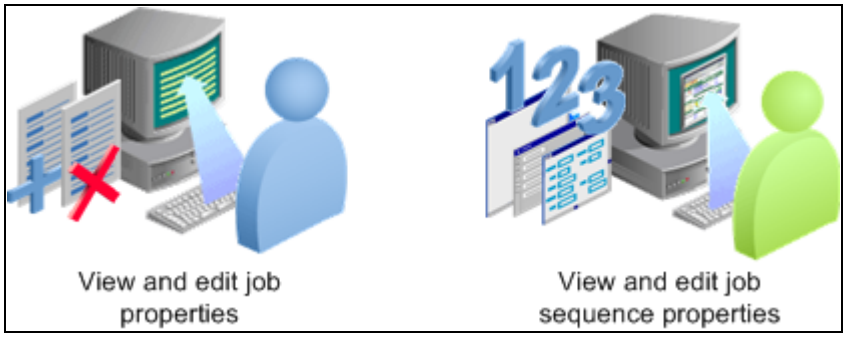


Job Run Options General Tab

You can also disable any message handlers specified for the job run in the General tab.

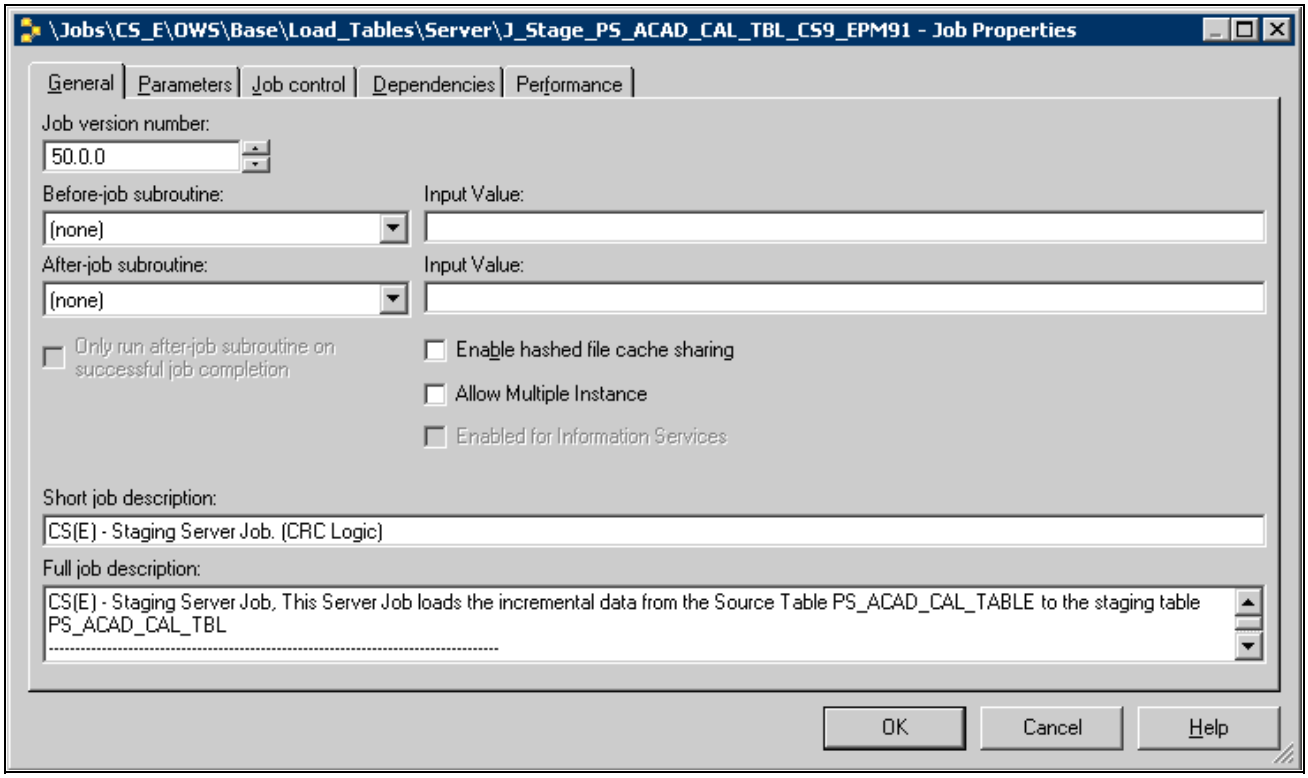
Editing Job Properties

The Job Properties window enables you to:



Editing Job Properties

To edit job properties, click the Job Properties button on the DataStage Designer toolbar.



Job Properties Window

The Job Properties window contains the following tabs:

Job Properties Page	*Description*
General	Enter name, category description, version number, before and after job subroutines, and their input values.
Parameters	Define parameters to represent processing variables. Operators can be prompted for values at run time.
Job Control	Set up a job control routine using BASIC functions to call and run other jobs from the current job. You can also set up job control by using the Sequence Editor in the Designer module.
Dependencies	Enter any dependencies that this job has on functions, routines, or other jobs.
Performance	Displays options for improving performance.

Using Expressions

Expressions define a value that is evaluated at run time.

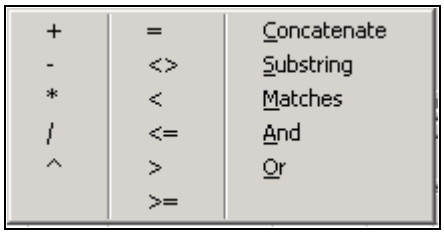
Simple expressions can contain:

- A string or numeric constant, for example, percent or 42.
- A variable name.
- A built-in or user-defined BASIC function.

A complex expression can contain a combination of constants, variables, operators, functions, and other expressions.

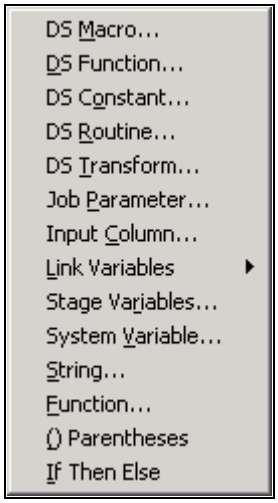
Accessing Expressions

You can access expressions by double-clicking a Transformer Stage within a job. Next you double-click the Derivation cell for any column in a transformer link and the Expression Editor opens. You can type an expression directly into the editor, or use the menu by clicking the Suggest button on the right side.



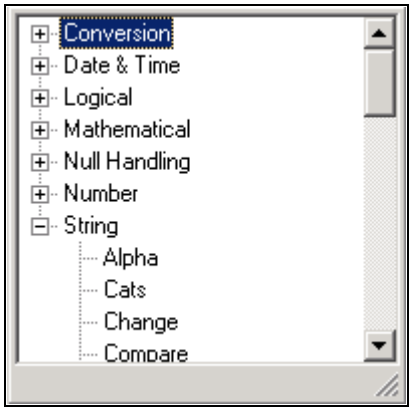
Expression Suggestion Menu

Different menus display depending on whether you right-click an input link, output link, the stage variable table, or links area background. The different menus are also dependent on what type of job you are working on (Server, Parallel, or Mainframe). The output link menu includes operations on Derivations. The input link menu includes operations on key expressions. The stage variable menu includes operations on stage variables.



Expression Editor Main Menu

To insert a function in your expression, click the Suggest button and select Function. The following choices are displayed:



Suggested functions

Click the plus sign next to the function type to display the functions categorized under them.

Expression Editor Options

This table describes the menu options for the Expression Editor:

Expression Editor Menu Options	***Description***
DS Macro	Insert a built-in DataStage BASIC Macro.
DS Function	Insert a built-in DataStage BASIC function.

Expression Editor Menu Options	*Description*
DS Constant	Insert a predefined constant.
DS Routine	Insert a routine from the repository.
DS Transform	Insert a transform from the repository.
Job Parameter	Insert a parameter.
Input Column	Insert an input column name.
Link Variables	Insert a link variable
Stage Variables	Insert a stage variable.
System Variables	Insert a system variable.
String	Insert a string.
Function	Insert a function.
() Parentheses	Insert parentheses.
If Then Else	Insert If Then Else logic into the expression.

Creating Constraints

Constraints, like expressions, enable you to filter or limit data based on criteria that you enter.

You can define a constraint or specify a reject link by both selecting an output link and clicking the *Edit constraints* button on the toolbar, or by double-clicking the output link header Constraint entry.

You can specify a constraint for each output link from a Transformer stage. You can also specify a particular link as a reject link. Reject links output rows that have not been written to any other output links from the Transformer stage.

In the example above, using the expression InsertFlag="Y" as a constraint verifies whether the lookup stages return a value. If no values were returned by the lookup stages, the InsertFlag field is set to N.

Using Hashed File Stages

Using hashed files improves job performance by enabling validation of incoming data rows without having to query a database each time a row is processed. These are called lookups. The hashed file can also be placed locally, eliminating time that would be spent accessing a remote server.

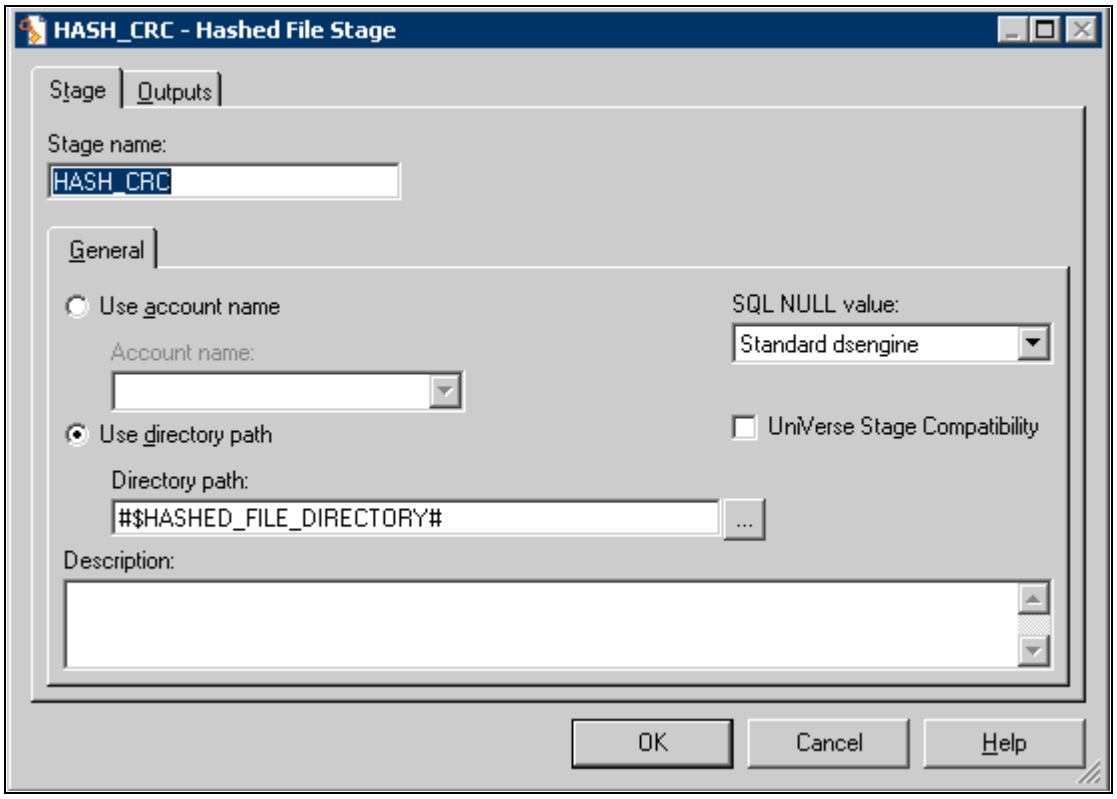
You can create hashed files to use as lookups in your jobs by running one of the delivered hash file jobs, or you can create a new job that creates a target hashed file. In many of the delivered PeopleSoft sequence jobs, the appropriate hashed file is refreshed as the last step following the load of the data table, which ensures synchronized updates to the data in the hashed file for use in future lookups.

Hashed file stages:

- Represent hashed files, which use a specific algorithm for distributing records in one or more groups, typically to store data extracted from a database.
- Can be used to extract or write data, or to act as an intermediate file in a job.
- Are most commonly used as reference tables or lookups based on key fields.
- Can have any number of inputs or outputs.
- Can be static or dynamic.

Accessing Hashed File Stages

To access a hashed file stage, double-click the hashed file stage in a job.



Hashed File Stage Properties Window

The Hashed File Stage window contains the following tabs:

Window Element	*Usage*
Stage tab	Define whether an account name or a directory path accesses the hashed file.
Inputs or Output tab	If the stage has an input, the Input tab is available. If it has an output, the Output tab is available.
Inputs - General tab	Enter the hashed file name. Select whether to create a new file, if none exists. Select whether to clear the file before writing to it.
Inputs - Columns tab	Select which columns of data will be written to the file.

Creating Hashed File Lookups

Lookups are references that enable you to compare each incoming row of data to a list of valid values, and then accept or reject that row based on the validation result.

DataStage job stages can have two types of input links:

- A Stream link represents where the data flow will flow, and is displayed as a solid line.
- A Reference link represents a table lookup, and is displayed as a dotted line.

Stream links, represented by solid lines, can connect either active or passive stages. Reference links, shown as dotted lines, are only used by active stages. Their purpose is to provide information that may affect how the data is changed, but they do not supply the actual data to be changed.

Typically, hashed files are used as lookups because they are much quicker to access than querying a database. Hashed files used as lookups usually contain only one or two key columns against which incoming data can be validated.

Before you can create the lookup, you must first create a hashed file containing the values to be used as a reference. To add a lookup stage to a job, you select the hashed file stage from the File palette, enter the directory path and file name of the hashed file, and link the hashed file stage to a transformer stage.

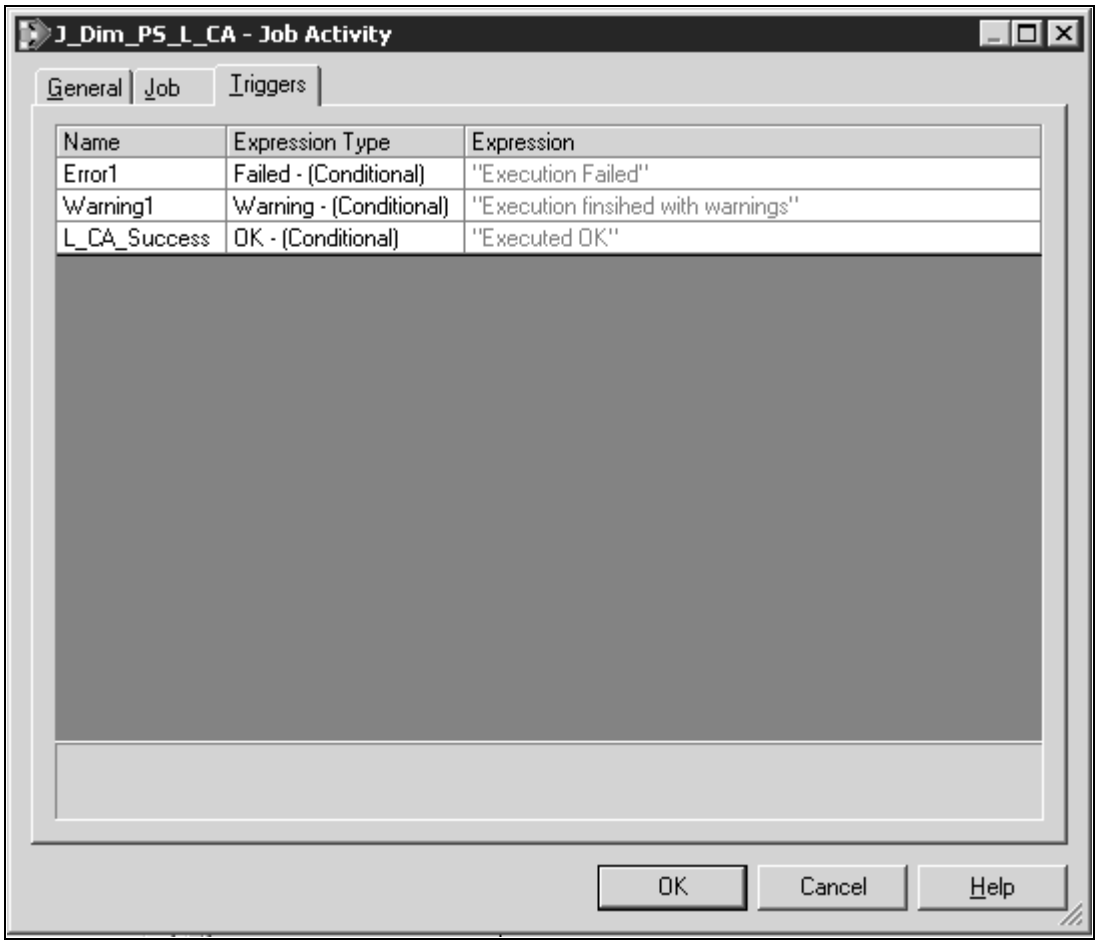
You use a DRS stage as a lookup when your lookup requires that use of relational operators, such as >= and <=.

Using Job Sequencers

Job Sequencers enable you to:

- Set up a sequence of server jobs to run.
- Specify control information such as different course of action depending on load type or other criteria.

Designing job sequencers is similar to designing server jobs. You create the job sequence in DataStage Designer, add activities from the palette, and then join or sequence activities together using links. You control the flow of the activity sequence using triggers.



Job Sequencer Trigger Window

Once you have defined a job sequence, you can schedule and run the job from DataStage Director.

Note. Job sequencers can also be run using DataStage Director.

Using DataStage BASIC

If you need to load data from a non-PeopleSoft source, you usually do not have to perform any programming tasks: you can use the delivered transforms and routines, using the delivered jobs and sequences as templates. For more complex jobs, you can use DataStage BASIC to:

- Define custom routines.
- Define custom transforms.
- Define derivation, expressions, and constraints in the transformer stage.

- Define before-job or after-job subroutines.
- Define job control routines.

PeopleSoft provides ETL jobs for loading data from PeopleSoft applications into EPM. Some of the PeopleSoft jobs use custom routines using DataStage BASIC.

Chapter 10

Using DataStage Director

This chapter provides an overview of DataStage Director and discusses how to:

- View Job Status
- Schedule Jobs
- Run Jobs
- Monitor Jobs
- Review Job Events

Note. This chapter does not discuss all the features available for DataStage Director. For a complete view of DataStage Designer functionality, please see the delivered IBM WebSphere documentation.

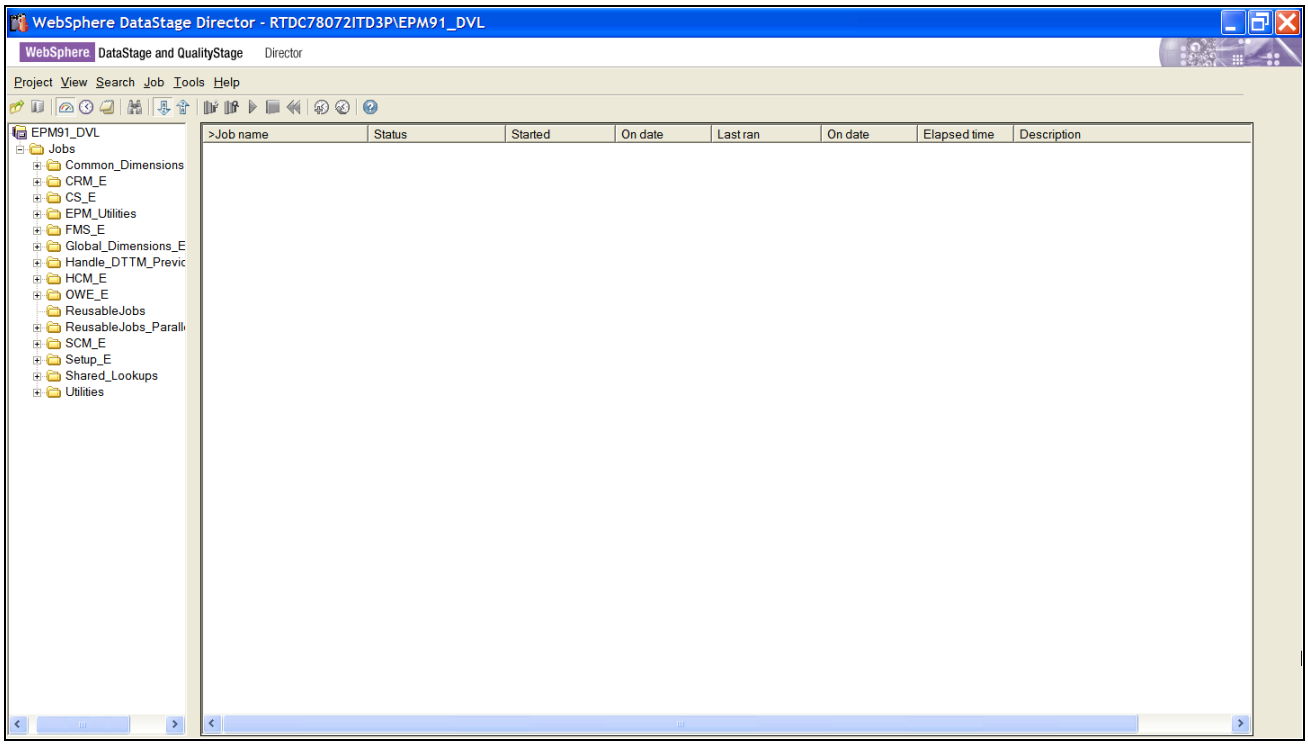
DataStage Director Overview

DataStage Director enables you to:

DataStage Director Properties

The DataStage Director window is divided into two panes:

- The Job Category pane lists all of the jobs in the repository.
- The right pane shows one of three views: Status view, Schedule view, or Log view.



DataStage Director Window

DataStage Director Menu Options

This table describes DataStage Director menu options:

Menu Option	***Description***
Project	Open another project, print, or exit.
View	Display or hide the toolbar, status bar, buttons, or job category pane, specify sorting order, change views, filter entries, show more details, or refresh the screen.
Search	Start a text search dialog box.
Job	Validate, run, schedule, stop, or reset a job, purge old entries from the job log file, delete unwanted jobs, clean up job resources (if this is enabled), set default job parameter values.
Tools	Monitor running jobs, manage job batches, start the DataStage Designer.
Help	Displays online help.

DataStage Director Toolbar Options

The following buttons appear on the DataStage Director toolbar:

DataStage Director Toolbar

This table describes the buttons provided with the DataStage Director toolbar

Toolbar Button	***Description***
Open	Open a project.
Print	Print the current list or log.
Status	Select the Job Status view.
Schedule	Select the Schedule view.
Log	Select the Job Log view.
Find	Search for a job.
Ascending	Sort the list in ascending order.
Descending	Sort the list in descending order.
Run Now	Run the currently selected job.
Stop	Stop the job run.
Reset	Reset the job and cancel any changes to the target.
Add to Schedule	Add a job to the schedule.
Reschedule	Change the job schedule.
Help	View online help.

DataStage Director View Options

DataStage Director has three view options:

- The Status view displays the status, date and time started, elapsed time, and other run information about each job in the selected repository category.
- The Schedule view displays job scheduling details.

- The Log view displays all of the events for a particular run of a job.

Viewing Job Status

The Job Status view in DataStage Director can display these possible states:

- *Compiled:* The job has been compiled, but not run since compilation.
- *Not compiled:* The job has not compiled successfully.
- *Running:* The job is currently running.
- *Finished:* The job has finished running.
- *Finished (see log):* The job has finished, but there are warning messages or rows were rejected.
- *Stopped:* The job was stopped by the operator.
- *Aborted:* The job finished prematurely.
- *Validated OK:* The job has been validated with no errors.

Scheduling Jobs

You can schedule a job to run in a number of ways:

- Once today at a specified time.
- Once tomorrow at a specified time.
- On a specific day and at a particular time.
- On the next occurrence of a particular date and time.
- Daily at a particular time.

To schedule a job In DataStage Director, select the Schedule button from the toolbar.

You can schedule each job to run on any number of occasions using different job parameters, if necessary.

Note. Microsoft Windows restricts job scheduling to administrators. You need to be logged in as an administrator to use the DataStage scheduling features.

Running Jobs

Each time that you run or schedule a job, you can:

- Change the job parameters and parameter values that are associated with the job.

- Override default limits for row processing and warning messages that are set for the job run.
- Set tracing options.

You set job options in the Job Run Options dialog box. They appear automatically when you start to run or schedule a job.

Setting Parameters

You can use the default parameter values, or enter another value. To reinstate the default values, click the Set to Defaultor All to Default button. Some job parameters, like dates, may be variables that you must enter for each job run.

If no parameters are set for a job, the system does not display the Parameters tab.

Setting Limits

Select the Limits tab to override any default limits for row processing and warning messages that are set for the job run. Click the Validate button to test the new settings or click the Run button to run the job.

Setting Tracing

Tracing helps analysts troubleshoot jobs. You can generate tracing information and performance statistics for server jobs.

The options on this page determine the amount of diagnostic information that is generated the next time a job is run. Diagnostic information is generated only for the active stages in a chosen job.

When the job runs, a file is created for each active stage in the job. The files are named using the format *jobname.stagename.trace*, and are stored in the &PH& subdirectory of your DataStage server installation directory.

Running, Stopping, Resetting, and Deleting Jobs

To run a job, select Job, Run Now, or click the Run button on the toolbar.

You can stop or reset a job. If a job is stopped or aborted, it might be difficult to trace where it ended. By resetting a job, you set it back to a state that can be run and, optionally, return your target files to their original state. If a job has aborted, it must be reset before you can run it again.

You can remove old or unwanted jobs from your project from either the Director or Designer.

Monitoring Jobs

Job monitoring enables you to review job progress through the links and to see the number of rows that are being processed.

To monitor a job In DataStage Director, select a job and then select Tools, New Monitor:

The Monitor window displays summary information about relevant stages in a job. It contains a tree structure that displays stages in a job and their associated links. For server jobs, it shows active stages. Active stages perform processing rather than reading or writing to a data source.

The Link type column displays up to four types of links:

Link Type	*Description*
<<Pri	Primary link
<<Ref	Reference link
<<Out	Output link
<<Rej	Reject link

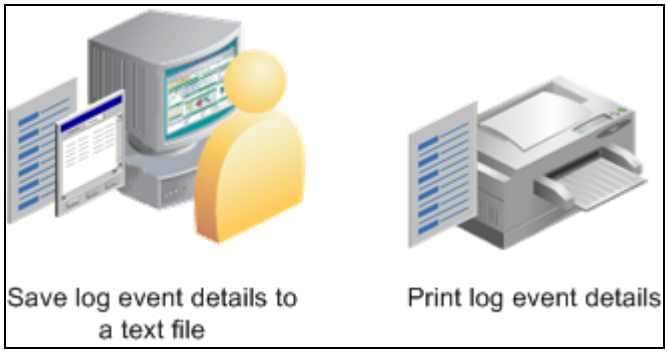
To see detailed information, double-click a link.

Setting the Refresh Interval

The Monitor window display is updated with new information from the server at regular intervals. You can set how often the updates occur by specifying a time, in seconds, in the Interval field. Click the arrow buttons to increase or decrease the value, or enter the value directly. The default setting is 10. The minimum value is 5. The maximum value is 65.

Reviewing Job Events

DataStage Director enables you to:



DataStage Director - Reviewing Job Events

You can filter:

- Jobs that appear in the Job Status view.
- Events that appear in the Job Log view.

Filtering Jobs

To filter jobs In DataStage Director, select View, Status, and then View Filter Entries.

This table describes job filtering options:

Window Element	Description
Include	Specify which jobs to include. Your choices are: • All jobs • Jobs matching *<a string criteria that you enter>* (case-sensitive)
Exclude	Specify which jobs to exclude. Your choices are: • No jobs • Jobs matching *<a string criteria that you enter>* (case sensitive)
Job Status	Your choices are: • All • All, except "Not compiled" • Terminated normally • Terminated abnormally
Released jobs	Include only released jobs.

Filtering Events

To organize log event details, you can filter events to select which log events are displayed, based on age, date/time, or event type. You can also specify the maximum number of entries that will appear.

To filter events In DataStage Director, select View, Log, then View, Filter Entries.

This table describes the events filtering options:

Window Element	Description
From:	Defines the earliest event to include: • Oldest. • Start of last run. • A specific date and time.

Window Element	*Description*
Through	Defines the most recent event to include: • Newest. • A specific date and time.
Limit	Limit the number of events that are displayed: • Select all entries. • Last *N* entries.
Type	Select one or more types of events that the system will display: • Information • Warning • Fatal • Reject • Other

Printing Jobs

To print jobs in DataStage Director, select Project, Print.

You can send the event log details either to your printer or to a text file.

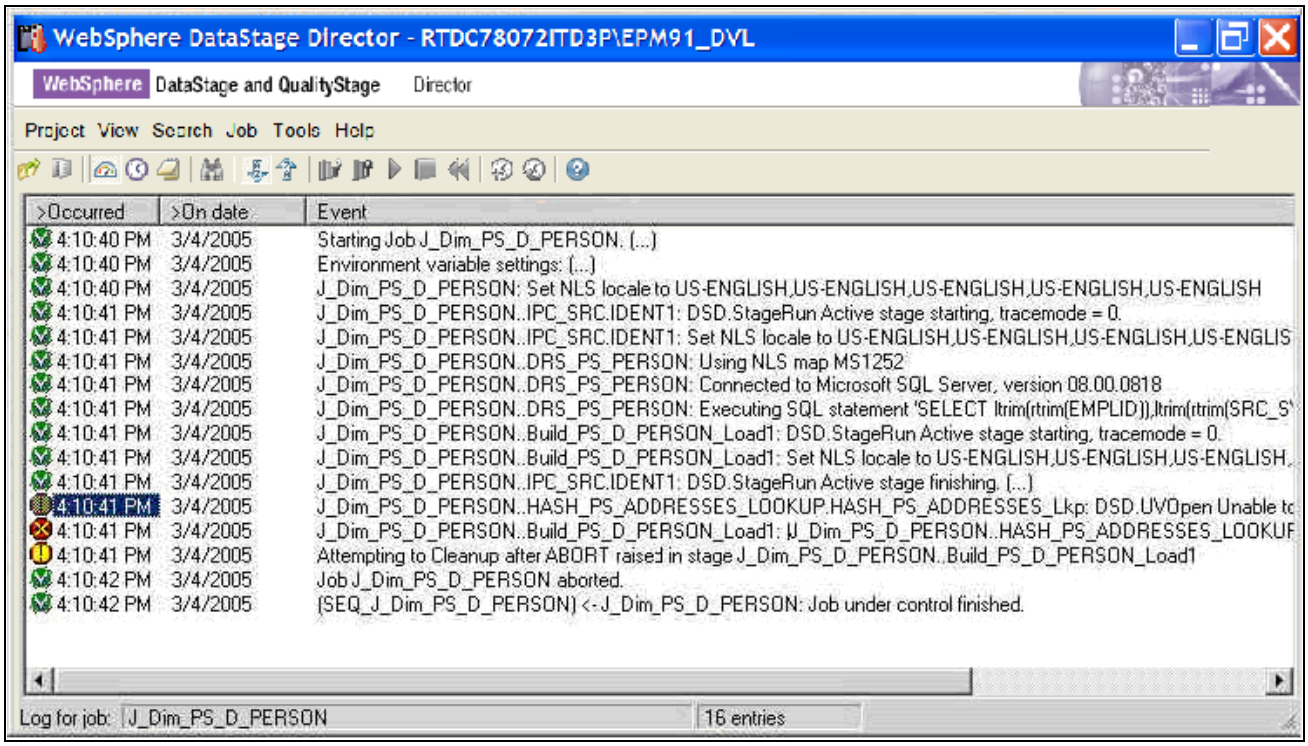
Managing Aborted and Failed Jobs

This section provides information on how you can manage aborted and failed jobs and discusses how to:

- Review the job log to determine job errors.
- Debug aborted and failed jobs.

Reviewing the Job Log to Determine Job Errors

The first step in managing aborted or failed jobs is to use DataStage Director to review the job log, which provides job run information.



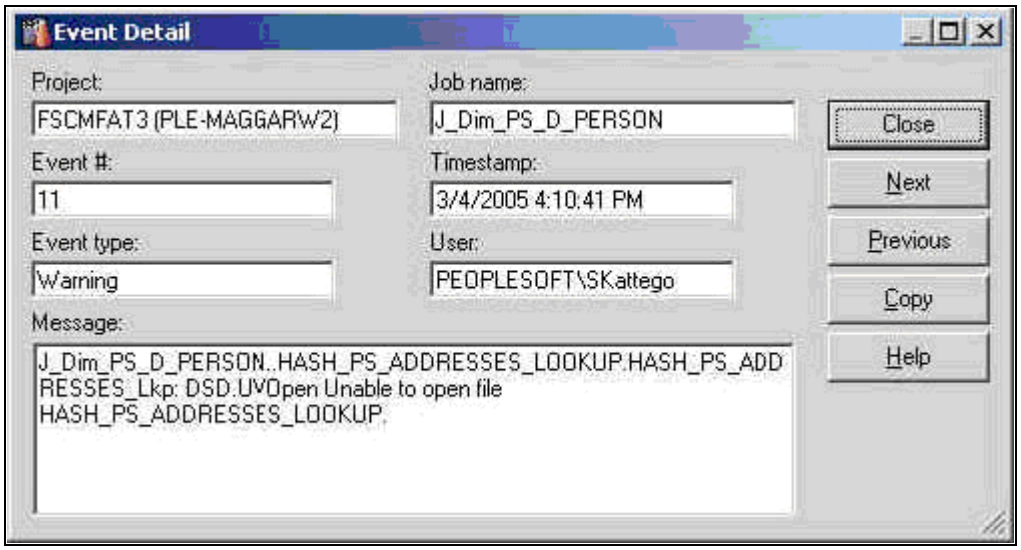
Detailed Job Log View of Aborted and Failed Jobs

Using the job log you can first determine which jobs require your attention. Note that the job log displays which jobs aborted or failed.

Jobs status are color-coded as follows:

- Green (V): Informational. Success condition.
- Yellow (I): Failed with warnings.
- Red (X): Error messages.

You can double-click an aborted or failed job to view details about the job.



Event Detail for Failed Job

A common cause for jobs aborting is that dependent hash files do not exist. This happens when a hash file that a job performs a lookup on has not been pre-created. The hash file load jobs have to be run. As you can see in the screen above, event details suggest that the job is missing the hash file HASH_PS_ADDRESSES_LOOKUP, which is required.

Debugging Aborted and Failed Jobs

Once you know which job has an issue, you can access the job in DataStage Designer and view the job with *performance statistics* on. This shows successful links in green and failed links in red, and helps target the specific part of the job design that failed. The performance statistics also gives the number of rows that have been transmitted through each link, again which information can be useful for debugging a job. Datastage Designer also provides advanced debugging features that can help developers set break points and watch variable values.

Jobs that run with more than on array size or transaction size usually result in a warning message; the job log displays a warning message relevant to each row of data in some instances. For example, if a job has a right string truncated error when inserting into the target database, the log specifically provides the row data that failed.

To address this type of issue, configure the job to limit the rows to process so there are less rows processed during job execution. This restricts the job run time and the log will also be smaller and more manageable.

Chapter 11

Setting Up DataStage for EPM

This chapter provides an overview of IBM WebSphere DataStage setup and discusses how to:

- Set up DataStage projects
- Configure environmental parameters
- Import *.dsx files

Understanding IBM WebSphere DataStage Setup and Configuration

This section provides an overview of DataStage setup and configuration.

Setting Up DataStage for EPM

In order for IBM WebSphere DataStage to work properly with EPM, you must perform the following setup and configuration steps:

- Create projects: Projects are used to import your *.dsx file metadata, which include ETL jobs.
- Setup and configure environmental parameters: Every ETL job uses environmental parameters and you must configure the default parameters.
- Import *.dsx files into projects: You must specify a project to hold your DataStage metadata, including ETL jobs.

Setting Up DataStage Projects

This section discusses considerations regarding the design of your DataStage projects and how to create a project.

Project Structure Considerations

Before you create your DataStage projects, you should consider how many projects you need to create. The number of projects you need depend on the following factors:

- The PeopleSoft source transaction system that you are using.

- The license codes for the Data Marts and/or EPM Applications you have purchased, or plan to implement.
- Disk space management based on storing hash files, log files, and *.dsx files.
- Common jobs necessary to all products (you may wish to create a separate project for the common jobs if you are implementing more than one data mart).

Based on the preceding information, you have the following options:

- Create one project per PeopleSoft source.

  For example, you can create a project called HCM_EPM90 for bringing data from an Enterprise – HCM source database.

- Create one project per EPM Warehouse (such as HCM or SCM warehouses).

  For example, you can create a project called CRM_EPM90 to handle all CRM-related Data Marts or EPM applications.

- Create one project for all the EPM Warehouses.

  For example, you can create a project called EPM90_ALL which will manage ETL jobs relating to all of the EPM Warehouses.

Because you need to create projects based on the preceding requirements, ensure that you have understood your long-term needs and requirements before creating suitable projects for importing the jobs.

Note. Certain operating systems, such as Unix, have a limit to the number of objects that can be created under one project or directory. Please check your operating system specifications before proceeding with project creation.

Note. There is no naming convention for projects.

Creating Projects

Use the following steps to create DataStage projects:

1. In the DataStage Administrator, enter the following information in the Attach to DataStage box:
   - *Domain:* Enter the name of the domain server machine.

     You can enter *localhost* if the client and server are installed on the same machine.

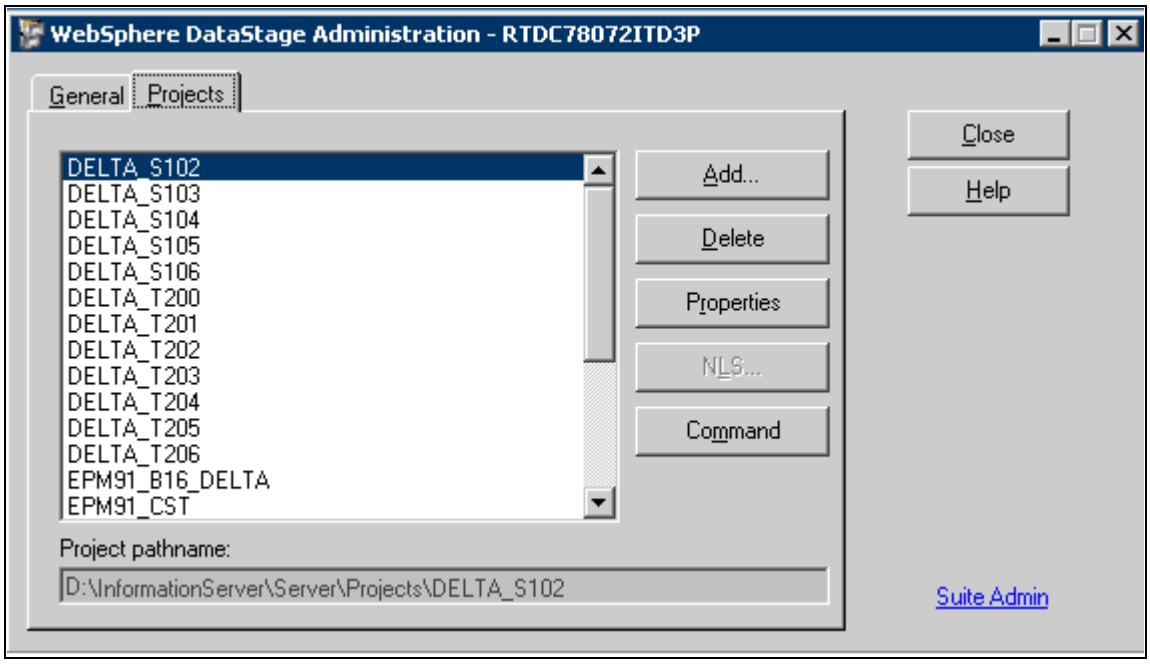
   - *User name:* Enter the user name that is required to log onto the machine on which the domain server is installed.

     The user name is the Windows login (if the server is running on a Windows platform) or the Administrator user that was used during the install (if the server is running on a Unix platform).

   - *Password:* Enter the password that is required to log onto the machine on which the domain server is installed.

   - *DataStage server:* Select the name of the server machine on which the DataStage engine is installed.
2. Click OK.

3. Click the Projects tab to view a list of available projects on the server.
4. Click the Add… button.



DataStage Administrator - Projects tab

The Add Project box displays.

5. Enter the project name in the *Name* field . You can specify your own naming convention in this field.

   Remember the project name is case-sensitive and you cannot change the project name after you have imported the dsx files. If you want to update or change the project, you only have the option to create a new project and import the dsx files again. DataStage Administrator automatically appends the project to the default location for a new project.

   Click the Browse button to select another location if the default location is not desirable (due to disk space constraints). Do not select the Create protected project check box as this would cause the project to be read-only.

6. Click OK to create the project on the server.

   This creates an empty project on the DataStage server. The updated list of projects displays after the copy is finished.

7. Repeat steps five through seven to create any additional projects.
8. When you are finished, click Close.

See Chapter 8, "Using DataStage Administrator," Setting Project Properties, page 192.

See Also

WebSphere DataStage Administration: Administrator Client Guide

Configuring Environmental Parameters

This section discusses how to copy the DSPARAMS file and add values for environmental variables. You must configure default environmental parameters for each project you are using, as every job uses a subset of the environmental variables.

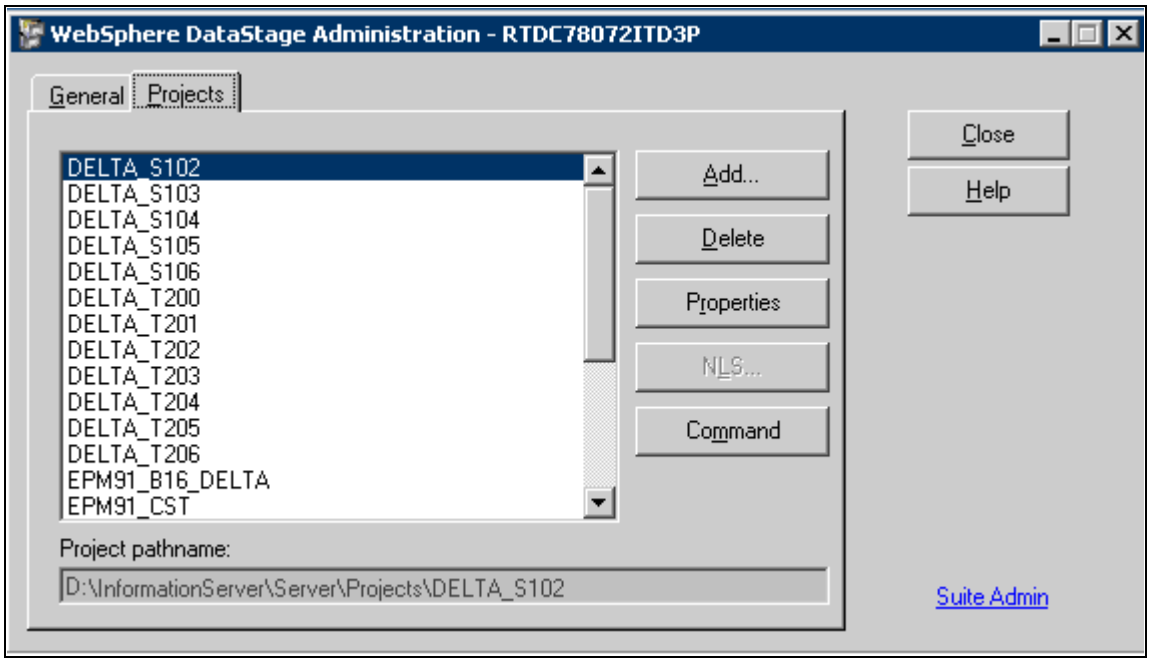
Copying the DSPARAMS File

Copying the DSPARAMS file enables you to bypass entering the parameters manually. Use the following steps to copy the DSPARAMS file:

1. Locate the DSPARAMS file using the following path on the installation CD:

   <PSHOME>\SRC\ETL.

2. In DataStage Administrator, navigate to the projects tab to determine the project home directory.



DataStage Administrator - Projects tab

   The project home directory path can be found under Project Pathname field.

3. Select your project name.

4. Use the project home directory path to navigate to your DSPARAMS file.

5. Rename the DSPARAMS file located in the project home directory folder to *DSPARAMS_old* and paste the copied file.

   Now, you can see a DSPARAMS file under this path.

6. Repeat steps one through five for each project you have created.

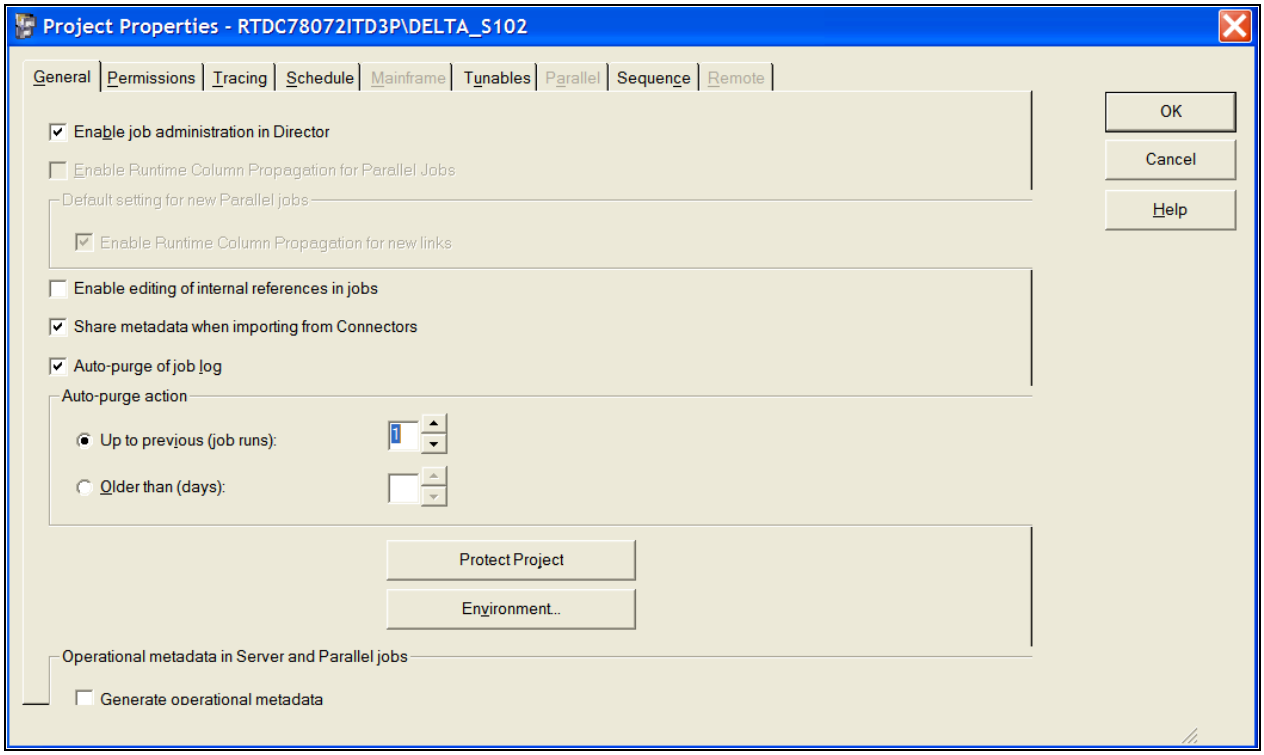
Adding Values for Environmental Variables

You must add suitable values for each environmental parameter you plan to use. For example, if you are only implementing an EPM Warehouse, you need not update OWE related parameters.

Use the following steps to add values for environmental variables:

1. In DataStage Administrator, select the Projects tab of the DataStage Administrator box to view a list of available projects on the server.
2. Select the project you would like to configure.
3. Click the Properties button.

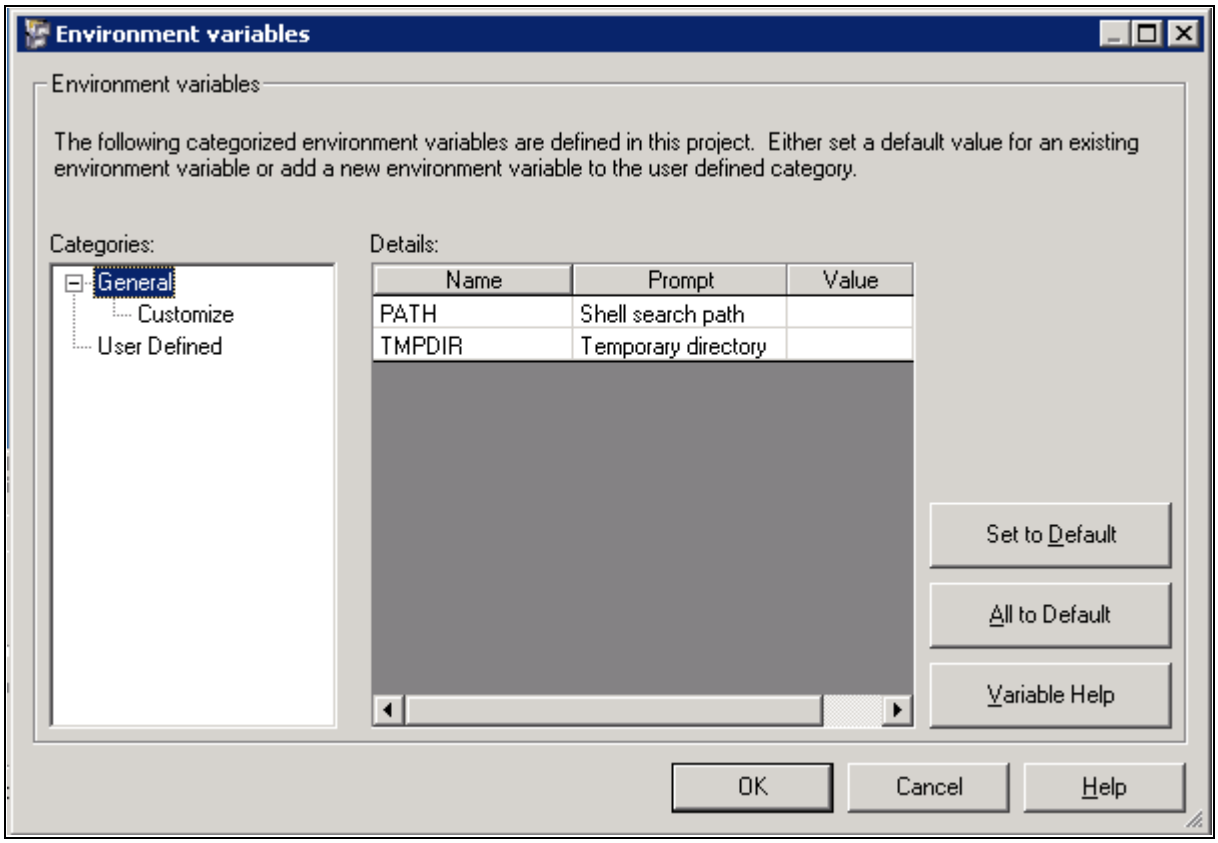
   The Project Properties window displays.



Project Properties - General tab

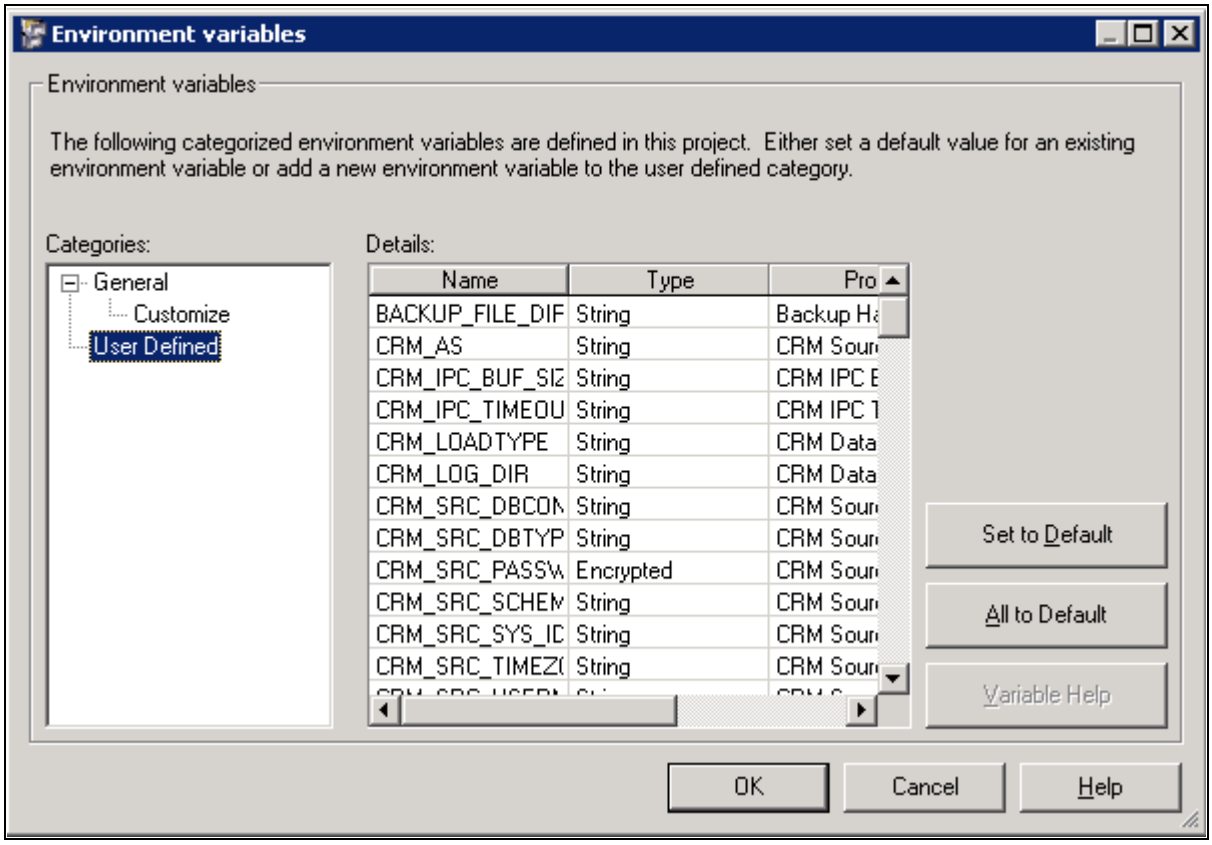
4. Click the Environment… button.

   The Environment Variables box displays.



Environment Variables window

5. In the Categories navigation frame, select *User Defined.*



Environment Variables window, User Defined variables selected

6. Scroll to the bottom of the list and enter the default values based on the warehouse setup.

   Ensure you do not include any trailing spaces in the values.

Note. Ensure that you perform a detailed analysis of the various environmental variables that impact your requirements with respect to the project created and the corresponding values. Use defaults wherever required. If you have not provided the correct values for the chosen environmental variables, the related jobs may abort or not work as required.

See Appendix C, "ETL Reference Documents," Environmental Parameters Information, page 671.

Importing Delivered .DSX Files

After you have created your projects and configured the project-level environmental parameters, you can import the .dsx files into the projects. PeopleSoft prepackaged .dsx files contain server jobs, job sequencers, and other metadata. The prepackaged jobs may belong to any of the following categories:

- Source to OWS
- Source to MDW (for CRM online marketing applications only)

- OWS to OWE
- OWS to MDW
- OWE to MDW

This section discusses preparations for importing .dsx files and how to import each type of .dsx file.

See Chapter 9, "Using DataStage Designer," Importing and Exporting Repository Components, page 215.

*Preparing to Import *.DSX Files*

Before you begin importing your *.dsx files, you should be thoroughly familiar with the *DSX Files Import Description* document located in the appendix of this guide.

See Appendix C, "ETL Reference Documents," DSX File Import Descriptions, page 671.

The *DSX Files Import Description* document contains the following information:

- The first column provides information on the order in which .dsx files should be imported.
- The second column indicates whether the .dsx file corresponds to a specific warehouse/SKU or generic file.
- The third column describes the type of jobs the .dsx file contains, such as OWS, OWE, and setup.
- The fourth column describes the source system associated with the .dsx file.
- The fifth column provides the name of the .dsx file.
- The sixth column provides the location of the .dsx file on the installation CD.
- The seventh column provides a detailed description of the .dsx file.

In order to import the .dsx files, you must be able to link to the Windows File Server from the IBM WebSphere DataStage Client.

Note. The .dsx files are delivered as zip files and must be unzipped before importing.

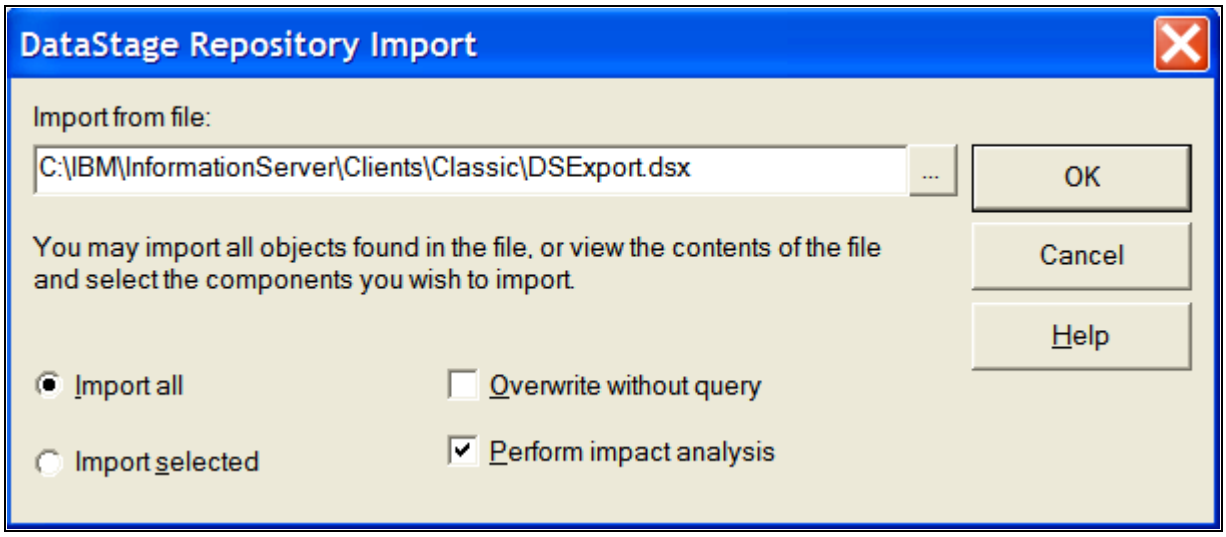
Note. The .dsx files can only be installed on a Microsoft Windows client operating system. If you have installed EPM on UNIX, you must access the files from the Windows file server.

Steps to Import a .DSX File

Use the following steps to import a .dsx file:

1. In DataStage Designer, select Import, DataStage Components from the menu.

   The DataStage Repository Import window displays.



   DataStage Repository Import Window

2. Click the Import from file browse button to locate the .dsx file you want to import.

   Do not select the Perform impact analysis check box unless you want to perform usage analysis. Selecting this check box increases the time it takes to import the .dsx files into a project.

3. Select Import All and click OK to import the file.

   Once the import process is complete, the Import Progress window closes.

Importing .DSX Files for EPM Analytical Applications Only

If you are planning to implement only the EPM Analytical Applications, the following sections provide a guide to .dsx files you need.

Import the following common .dsx files:

- COMMON_UTILITIES.dsx
- COMMON.dsx
- OWE.dsx

Import any or all of the following .dsx files, depending upon the number of PeopleSoft source transaction systems that you may have:

- WCS_OWS.dsx (For a PeopleSoft Campus Solutions database)
- WCR_OWS.dsx (For a PeopleSoft CRM source database)
- WFN_OWS.dsx (For a PeopleSoft FMS source database)
- WHR_OWS.dsx (For PeopleSoft HRMS and Learning Management source database)

- WSC_OWS.dsx (For a PeopleSoft SCM source database)

See Appendix C, "ETL Reference Documents," DSX File Import Descriptions, page 671.

Importing .DSX Files for EPM Warehouses Only

If you are planning to implement only EPM Warehouses, import the following common .dsx files:

- COMMON_UTILITIES.dsx
- COMMON.dsx

Each EPM Warehouse (for example, Campus Solutions Warehouse or HCM Warehouse) requires a unique list of *.dsx files to import.

The appendix *ETL Reference Documents* helps you determine which .dsx files you need to import for your warehouse.

See Appendix C, "ETL Reference Documents," DSX File Import Descriptions, page 671.

Chapter 12

Defining ETL Parameters

Certain EPM products require that you specify input parameters for ETL jobs prior to running the jobs. For example, some parameters specify row-selection criteria and load methodologies.

This chapter discusses how to define ETL parameters for:

- Planning and Budgeting analytical application
- EPM Warehouses (FMS Warehouse and CRM Warehouse)

Defining ETL Parameters for the Planning and Budgeting Analytical Application

This section discusses how to define financial asset item parameters for the Planning and Budgeting analytical application.

Page Used to Define Financial Asset Item Parameters

Page Name	*Definition Name*	*Navigation*	*Usage*
Financial Asset Item Setup	PF_FIN_AST_SETUP	EPM Foundation, EPM Setup, ETL Map Parameters, Financial Asset Item Setup, Financial Asset Item Setup	Enter the setID for the rows to be deleted in the OWE table PS_BP_ASSET_ITEMS and reloaded from BD_ASSET_ITEMS.

Setting Up Financial Asset Item

Access the Financial Asset Item Setup page (EPM Foundation, EPM Setup, ETL Map Parameters, Financial Asset Item Setup, Financial Asset Item Setup).

Financial Asset Item Setup page

SetID Enter the setID associated with the asset data.

Del. Flag (Y/N) [Delete Flag (Y/N)] Check to indicate that the rows in the target table PS_BP_ASSET_ITEMS that match the setID on this page are to be deleted.

Defining ETL Parameters for the EPM Warehouses

This section describes how to set ETL parameters for data marts within the FMS and CRM warehouses.

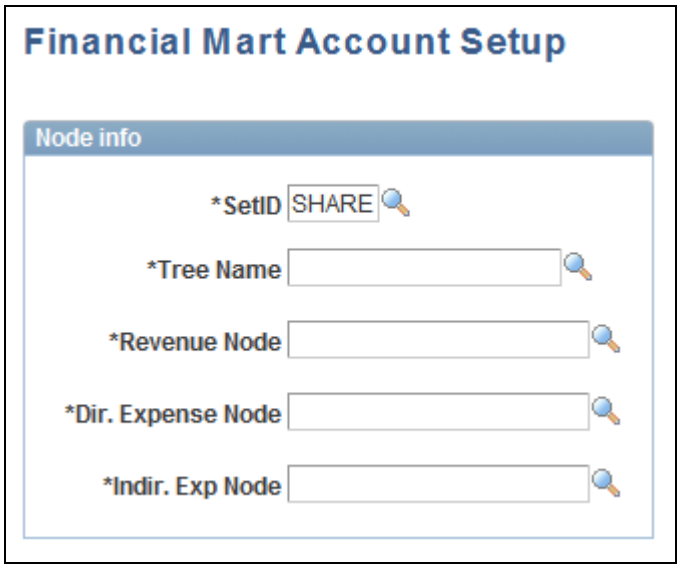
Pages Used to Define ETL Parameters

Page Name	*Definition Name*	*Navigation*	*Usage*
Financial Mart Account Setup	PF_FIN_ACCT_SETUP	EPM Foundation, EPM Setup, ETL Map Parameters, Financial Mart Account Setup, Financial Mart Account Setup	Enter tree and node details that represent accounts on which Profitability analysis is performed.
Financial Mart GC Cled Setup	PF_FIN_CLED_SETUP	EPM Foundation, EPM Setup, ETL Map Parameters, Financial Mart GC Cled Setup, Financial Mart GC Cled Setup	Enter values such as Business Unit, Scenario, and Period that determine the rows to be deleted from Multidimensional Warehouse table PS_F_CLEDGER and reloaded from PS_GC_CLED_MGT_F00. in the MDW table F_CLEDGER.

Page Name	*Definition Name*	*Navigation*	*Usage*
Financial Mart GC Flow Setup	PF_FIN_FLOW_SETUP	EPM Foundation, EPM Setup, ETL Map Parameters, Financial Mart GC Flow Setup, Financial Mart GC Flow Setup	Enter values such as Business Unit, Scenario, and Period that determine the rows to be deleted from MDW table PS_F_FLOWS and reloaded from PS_GC_FLOW_MGT_F00
Financial Rolling Average Setup	PF_FIN_RAVG_SETUP	EPM Foundation, EPM Setup, ETL Map Parameters, Financial Rolling Avg. Setup, Financial Rolling Average Setup	Enter values that are used to determine rolling averages for loading PS_F_LEDGER.
AP Mart Aging Setup	PF_FIN_AP_SETUP	EPM Foundation, EPM Setup, ETL Map Parameters, AP Mart Aging Setup, AP Mart Aging Setup	Enter values that are used in calculating the AP Aging process.
AR Mart Aging Setup	PF_FIN_AR_SETUP	EPM Foundation, EPM Setup, ETL Map Parameters, AR Mart Aging Setup, AR Mart Aging Setup	Enter values that are used in calculating the AR Aging process.
Customer Segment Ranking Map	PF_CUSEG_RANK_MAP	EPM Foundation, EPM Setup, ETL Map Parameters, Customer Segment Rank Map, Customer Segment Ranking Map	Force rank customer segments.

Defining Account Parameters for the GL and Profitability Data Mart

Access the Financial Mart Account Setup page (EPM Foundation, EPM Setup, ETL Map Parameters, Financial Mart Account Setup, Financial Mart Account Setup).



Financial Mart Account Setup page

Use this page to enter details of nodes that belong to the Account tree. Only accounts created under these nodes are loaded by the ETL process into the PS_F_PROFITABILITY table.

SetID Enter the setID for the source data.

Tree Name Enter the name of the Account tree.

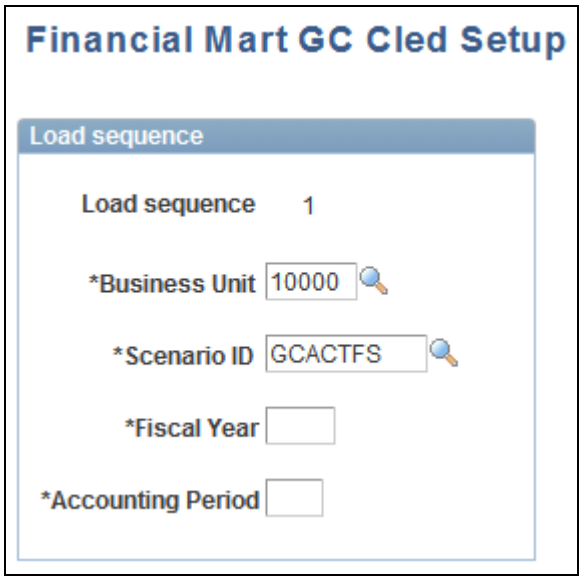
Revenue Node Enter the Account tree node that relates to revenue accounts.

Dir. Expense Node (Direct Expense Node) Enter the Account tree node that relates to direct expense accounts.

Indir. Exp Node (Indirect Expense Node) Enter the Account tree node that relates to indirect expense accounts.

Defining Global Consolidations C-Ledger Parameters for the GL and Profitability Data Mart

Access the Financial Mart GC Cled Setup page (EPM Foundation, EPM Setup, ETL Map Parameters, Financial Mart GC Cled Setup, Financial Mart GC Cled Setup).



Financial Mart GC Cled Setup page

The sequence number on this page indicates that the process accepts multiple sets of business unit, scenario, fiscal year, and accounting period as input. The ETL job processes each set of input data in the sequence indicated.

Load Sequence Enter the sequence in which the ETL job processes the input parameters. The ETL process accepts multiple sets of business unit, scenario, fiscal year and accounting period as input. The ETL job processes each set of input data in the sequence indicated.

Business Unit Enter the performance business unit.

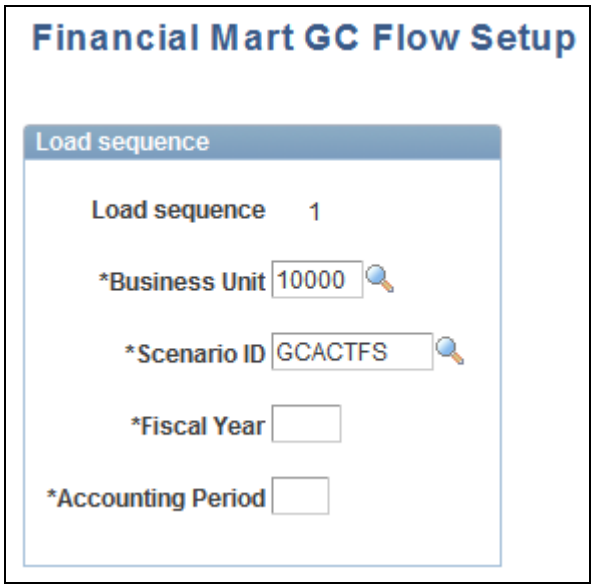
Scenario ID Enter the scenario ID.

Fiscal Year Enter the fiscal year.

Accounting Period Enter the accounting period.

Defining Global Consolidations Flow Parameters for the GL and Profitability Data Mart

Access the Financial Mart GC Flow Setup page (EPM Foundation, EPM Setup, ETL Map Parameters, Financial Mart GC Flow Setup, Financial Mart GC Flow Setup).



Financial Mart GC Flow Setup page

Load Sequence Enter the sequence in which the ETL job processes the input parameters. The ETL process accepts multiple sets of business unit, scenario, fiscal year and accounting period as input. The ETL job processes each set of input data in the sequence indicated.

Business Unit Enter the performance business unit.

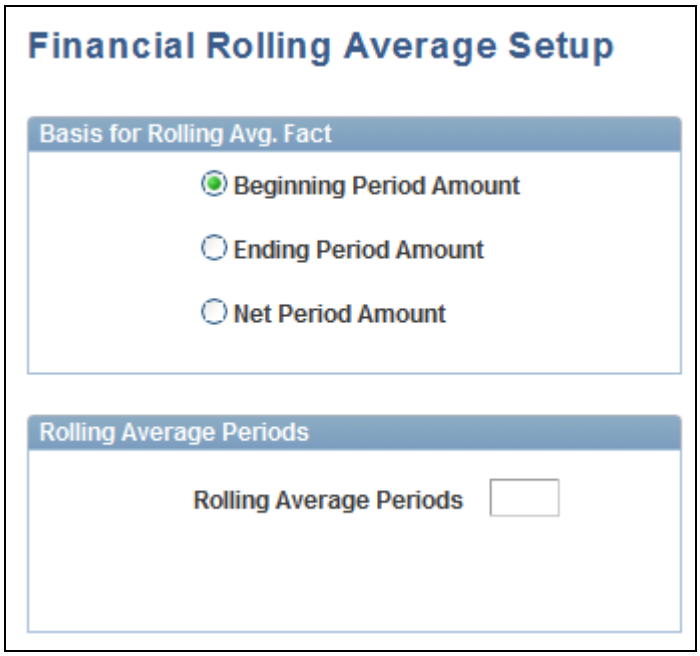
Scenario ID Enter the scenario ID.

Fiscal Year Enter the fiscal year.

Accounting Period Enter the accounting period.

Defining Financial Rolling Average Parameters for the GL and Profitability Data Mart

Access the Financial Rolling Average Setup page (EPM Foundation, EPM Setup, ETL Map Parameters, Financial Rolling Avg. Setup, Financial Rolling Average Setup).



Financial Rolling Average Setup page

Basis for Rolling Avg. Fact (Basis for Rolling Average Fact) Select the basis for the Rolling Average fact. Values are:

Beginning Period Amount: The ETL process uses the Beginning Amount from the PS_F_LEDGER table to calculate the rolling average.

Ending Period Amount: The ETL process uses the End Amount from the PS_F_LEDGER table to calculate the rolling average.

Net Period Amount: The ETL process uses the Net Amount from the PS_F_LEDGER table to calculate the rolling average.

Rolling Average Periods Enter the number of periods used in calculating the average value.

Defining Aging Parameters for the Payables Data Mart

Access the AP Mart Aging Setup page (EPM Foundation, EPM Setup, ETL Map Parameters, AP Mart Aging Setup, AP Mart Aging Setup).

SetID Enter the setID associated with data to be used for the Aging process

Aging ID Enter the aging ID, for example *MONTH.*

Calendar ID Enter the calendar ID.

Accounting Period Enter the accounting period.

Incl Draft flag (Include Draft flag) Check to include drafts in AP Aging process.

Aging Date Type Enter the Aging Date Type. Values are:

A: The application uses the Accounting Date.

T: The applications uses Invoice Date.

Fiscal Year Enter the fiscal year.

Incl PrePay flag (Include PrePay flag) Enter the value to indicate whether to include prepay in the AP Aging process. Values are:

Y: Include prepay.

N: Do not include prepay.

Unrecord Liab Ind (Unrecorded Liability Indicator) Enter the value to indicate whether to include unrecorded liability in the AP Aging process Values are:

Y: Include unrecorded liability.

N: Do not include unrecorded liability.

Effective Date Enter the effective date of the record.

Source System Identification Enter the name of the source system.

Defining Aging Parameters for the Receivables Data Mart

Access the AR Mart Aging Setup page (EPM Foundation, EPM Setup, ETL Map Parameters, AR Mart Aging Setup, AR Mart Aging Setup).

SetID Enter the setID for the data used for the AR Aging process.

Aging ID Enter the number of days used for days sales outstanding (DSO) in the form of a band, for example *30–60.*

Fiscal Year Enter the fiscal year.

Incl Grace Days (Include Grace Days) Enter the value to indicate whether to include grace days in the AR Aging process Values are:

Y: Include grace days.

N: Do not include grace days.

DSO Fiscal Year (Days Sales Outstanding Fiscal Year) Enter the fiscal year used as input to DSO.

Bad Debt Reason Enter the reason code for bad debt.

Tot AR Specialist (Total Accounts Receivable Specialist) Enter the number of AR specialists.

Effective Date Enter the effective date of the record.

Accounting Period Enter the accounting period.

Sales History ID Enter the customer sales history ID.

DSO Acct Period (Days Sales Outstanding Accounting Period) Enter the accounting period used as input to DSO.

Tot Credit Analyst (Total Credit Analyst) Enter the number of credit analysts.

Tot Collect Analyst (Total Collect Analyst) Enter the number of collection analysts.

Source System Identification Enter the name of the source system.

Defining Customer Segment Ranking Parameters for the Customer Segment Data Mart

If you perform customer segment analysis, you may need to define parameters to set up customer segment ranking prior to moving data into the MDW. You must do this if you plan to populate the Segment dimension and perform customer segment ranking. The Segment dimension source the marketing tables RA_LIST and RA_LIST_REC, which come from the OWS.

If the OWS tables RA_LIST and RA_LIST_REC are populated and you plan to analyze customers by segment, you must rank customers on the Customer Segment Ranking Map page before you populate the Segment dimension in the MDW. If you have not populated these source marketing tables, you do not need to set up the customer ranking parameters.

Accessing the Customer Segment Ranking Map page

Access the Customer Segment Ranking Map page (EPM Foundation, EPM Setup, ETL Map Parameters, Customer Segment Rank Map, Customer Segment Ranking Map).

Use this page to enter the numeric ranking for each customer segment. You must use one unique ranking for each segment. If a customer belongs to more than one segment within a segment group, the ETL process puts that customer in the highest ranking (lowest number) segment within the selected segment group.

If you do not find any values on the search page for the Customer Segment Ranking Map page, you do not need to use this page; that is, there are no customer segments for you to rank.

To retrieve data to populate this page, run the OWS ETL jobs J_STAGE_PS_RA_LIST and J_STAGE_PS_RA_LIST_REC.

SetID Displays the setID for the customer segment group.

Source ID Displays the source ID in the source database.

Segment Group Displays the identifier for the segment group.

Segment Displays the identifier for the segment in the segment group.

Note. The first time you populate the Segment dimension, you are prompted to enter a segment group value. This segment group value remains, unless you must change the value for subsequent runs of the ETL job to populate the Segment dimension. To change the value of the segment group, in DataStage Designer, click the job property J_DIM_PS_P_SEGMENT, click the Parameter tab, and enter the new value for Segment_Group.

Name Displays the segment name.

Rank Enter the unique numeric ranking of the segment within the segment group.

Chapter 13

Running Initial Setup Jobs

Initial setup jobs are a group of jobs that you run when you initially populate target OWS tables with data and need to set up common ETL components. These jobs set up your hashed files, shared lookups, and common dimensions, and bring PeopleSoft operational source data into the OWS tables. These jobs are common to all EPM products.

This chapter discusses how to:

- Verify ETL components have been imported properly
- Compile ETL jobs
- Run ETL setup jobs to bring source data into EPM
- Run Shared Lookup Jobs
- Run Setup - OWE Jobs
- Run Common Dimension Jobs

Note. Running initial setup jobs are required for both the EPM Warehouses and the Analytical Applications. However, additional implementation jobs are required to set up the EPM Warehouses and the Analytical Applications:

see the chapter entitled 'Importing Source Business Units into EPM to Create Warehouse Business Units' in this PeopleBook, for both the EPM Warehouses and the Analytical Applications.

see the chapter entitled 'Running the [*product name*] Warehouse Implementation Jobs' in your specific EPM warehouse PeopleBook (for example, Running the HCM Warehouse Implementation Jobs in the HCM Warehouse PeopleBook).

Verifying ETL Components Have Imported Properly

After you have finished configuring DataStage for EPM and imported all of the appropriate *.dsx files (which include different ETL components) you must verify that all the necessary components have been imported properly. This must be performed prior to running any ETL setup jobs.

Verifying Routines

Perform the following steps to verify that your ETL routines are present:

1. In DataStage Designer, attach to your project and expand the Routines node in the left navigation panel of the window.
2. Verify that the object, EPM90_Routines, is present in the list of routines.

   If this object does not exist in the list, your import of the Common_Utilities.dsx file was unsuccessful. You must re-import the *.dsx file.

Verifying Shared Containers

Perform the following steps to verify that your shared containers are present:

1. In DataStage Designer, attach to your project and expand the Shared Containers node in the left navigation panel of the window.
2. Verify that the objects, Incremental_Logic and Language_Swap, are present in the list of shared containers. The Incremental_Logic object should also contain six components and Language_Swap should contain one.

   If these objects do not exist in the list, your import of the Common_Utilities.dsx file was unsuccessful. You must re-import the *.dsx file.

Verifying ETL Jobs

Perform the following steps to verify that your ETL jobs are present:

1. In DataStage Designer, attach to your project and expand the Jobs node in the left navigation panel of the window.
2. Expand each of the sub-folders in the Jobs node, such as Common_Dimensions,Global_Dimensions_E, and Shared_Lookups, and verify that each folder has the requisite ETL jobs in it.

   The number of jobs present in each sub-folder vary depending on the product you are implementing.
3. Repeat the first two steps for each product and related project (for example HCM Warehouse).

Compiling ETL Jobs

Before you run any ETL setup jobs, you must compile all jobs first. The jobs should be compiled after you imported the related *.dsx file. The following sections discuss how to verify if your jobs are compiled, and compile those that might not have been.

Verifying ETL Job Compilation

Perform the following steps to verify that your ETL jobs have been properly compiled:

1. In DataStage Director attach to your project and select *View, Status* from the menu.

2. In the left navigation panel of the DataStage Director window, expand the Jobs node.

   Verify that the status of all jobs are equal to *compiled.*

   If any of the jobs are not compiled, compile them using the steps outlined in the following sections.

Compiling Individual ETL Jobs

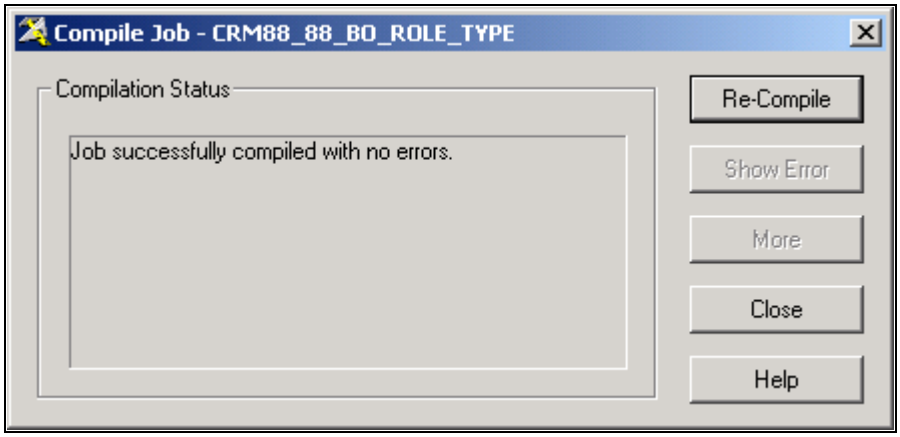
Perform the following steps to compile individual ETL jobs:

1. In DataStage Designer, navigate to the job you want to compile, open it, and click on the Compile button.

   Compile Button

   After compiling the job you receive a message informing you of the outcome in the Compilation Status window.

   Compilation Status Window

2. If the job compiled with no errors, click Close.

   If the job compiled errors, click Re-Compile.

3. Repeat steps one and two for each job you wish to compile.

Compiling Multiple ETL Jobs

Perform the following steps to compile multiple ETL jobs:

1. In the DataStage Designer attach to your project and select *Tools, Run Multiple Job Compile* from the menù`.

   The *DataStage Batch Job Compilation Wizard* opens.

2. In the wizard, select the Server, Sequence, Only select uncompiled jobs, and Show job selection page check boxes.

3. The right panel of the wizard window lists all uncompiled jobs.

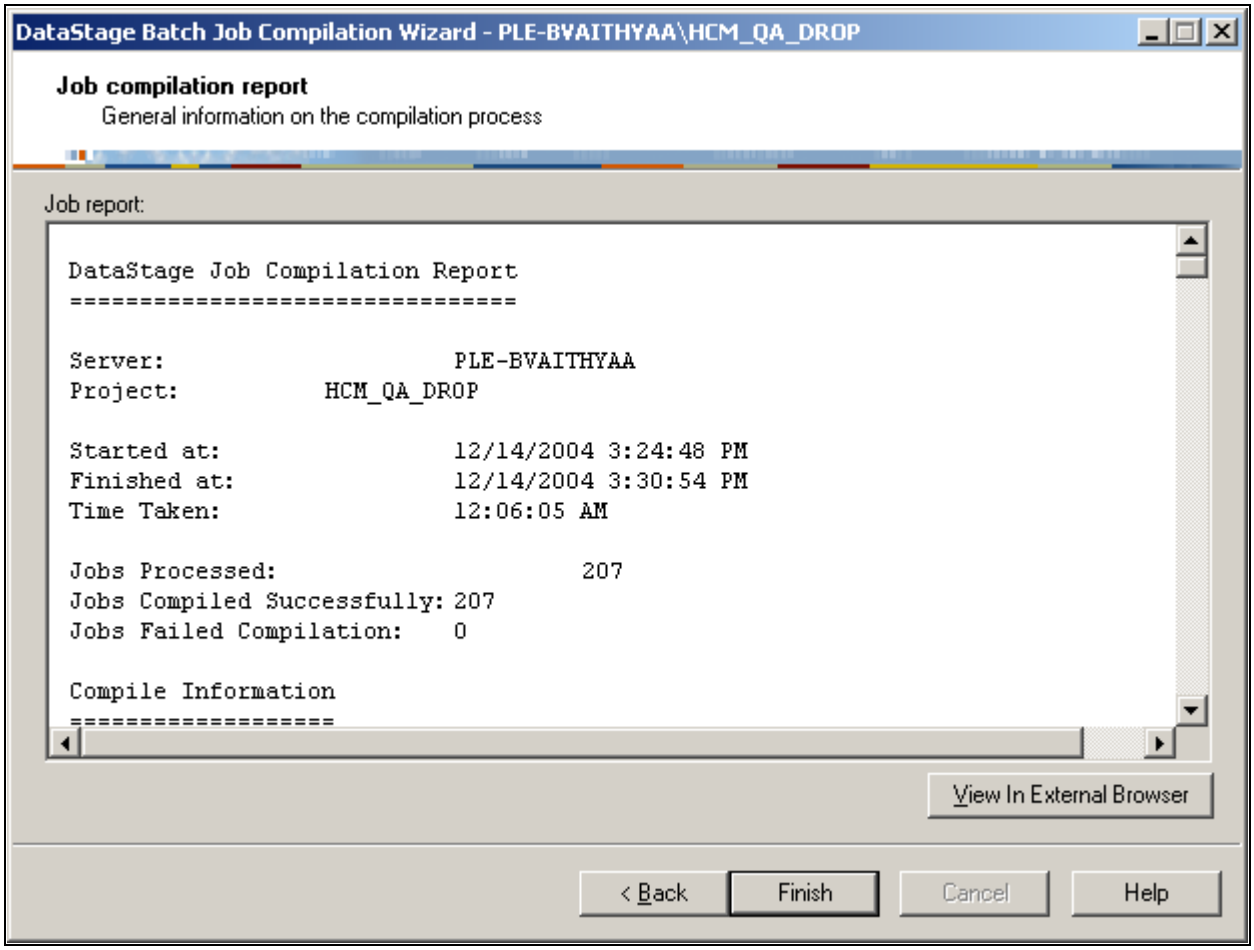
   Click Next.

4. Click the Start Compile button.

   After job compilation is complete, the status for each job reads *Compiled OK*.

5. Click Next, then Finish to complete the process.

   The Job Compilation Report displays for you to review, should you wish to do so.



Job Compilation Report

See Also

WebSphere DataStage Development: Designer Client Guide

Running ETL Setup Jobs to Bring Source Data Into EPM

After you verify that all ETL components have been successfully imported and all ETL jobs compiled, you are ready to run the jobs which bring your source data into the EPM database (the OWS Load_Hash_Files and Load_Tables jobs).

You have the option of running these jobs manually or using the Master Run Utility.

To run the jobs automatically with the Master Run Utility, follow the steps provided in the ETL Configurations chapter of this book.

To run the jobs manually, follow the steps described below.

Running Hash Files Setup Jobs Manually

Perform the following steps to manually run hash files setup jobs:

1. In DataStage Director, navigate to the hash file jobs by expanding the nodes in the left navigation panel using the following path: *Setup_E, OWS, <Warehouse Code>,Base, Load_Hash_Files, Server.*

   Note. *Warehouse Code* refers to each of the EPM Warehouse products (for example CS Warehouse or HCM Warehouse).

2. Select each hash file setup job in the Job Status view and select *Job, Run Now...* from the menu.

   The Job Run Options box appears.

3. Update the job parameters if necessary and click Run.

   The job is scheduled to run with the current date and time, and the job's status is updated to *Running.*

Running the Setup - OWS Jobs Manually

Perform the following steps to manually run setup - OWS jobs:

1. In DataStage Director, navigate to the setup jobs by expanding the nodes in the left navigation panel using the following path: *Setup_E, OWS, <Warehouse Code>, Base, Load_Tables, Sequence.*

   Note. *Warehouse Code* refers to each of the EPM Warehouse products (for example, CS Warehouse or HCM Warehouse).

2. Select each setup - OWS job in the Job Status view and select *Job, Run Now...* from the menu.

   The Job Run Options box appears.

3. Update the job parameters if necessary and click Run.

   The job is scheduled to run with the current date and time, and the job's status is updated to *Running.*

Running Shared Lookup Jobs

Shared lookups function the same as hash file lookups—they act as views of specific EPM warehouse tables and contain only a subset of the data available in a warehouse table. These streamlined versions of warehouse tables are used to perform data validation (lookups) within an ETL job and select specific data from lookup tables (such as sourceID fields in dimensions). The only difference between a regular lookup and a shared lookup is that the shared lookups are used across all EPM products.

Because shared lookups are essential in the lookup process, jobs cannot function properly until all hash files are created and populated with data. Before you run any job that requires a hash file, you must first run all jobs that create and load the hash files—also called initial hash file load jobs.

Steps Required to Run Shared Lookup Jobs

Perform the following steps to run the shared lookup jobs:

1. In DataStage Designer, attach to your project and expand the Shared_Lookups node in the left navigation panel of the window.

   The following sub-folders exist in the Shared_Lookups node:

   - Control_Tables
   - DimensionMapper_Lookups
   - Language_Lookups
   - System_Lookups

2. Select one of the sub-folders.
3. Select the lookup jobs in the Job Status view and select *Job, Run Now...* from the menu.

   The Job Run Options box appears.

4. Update the job parameters if necessary and click Run.

   The job is scheduled to run with the current date and time, and the job's status is updated to *Running.*

5. Repeat steps two and three for the remaining sub-folders.

Running Setup - OWE Jobs

Setup - OWE jobs load the setup tables used in standard OWE jobs (jobs that move your operational data from the OWS to the OWE). You can run these jobs manually or use the Master Run Utility. To run the jobs automatically with the Master Run Utility, follow the steps provided in the ETL Configurations chapter of this book.

Perform the following steps to run the setup - OWE jobs manually:

1. In DataStage Director, navigate to the setup OWE jobs by expanding the nodes in the left navigation panel using the following path: *Setup_E, OWE, Base, Load_Tables, Sequence.*
2. Select each setup - OWE sequence job in the Job Status view and select *Job, Run Now...* from the menu.

   The Job Run Options box appears.
3. Update the job parameters if necessary and click Run.

   The job is scheduled to run with the current date and time, and the job's status is updated to *Running.*

See Chapter 15, "ETL Configurations," Using the Master Run Utility to Automatically Run Your ETL Jobs, page 321.

Running Common Dimension Jobs

Common dimensions are dimensions that are shared across *all* EPM products. Not only do these dimensions play an important role in all reporting and analytical analysis, but they are particularly important to the Allocation Manager data enrichment tool, used by EPM Analytical Applications. In Allocation Manager, these dimensions are used to determine the divisor, therefore the ratio, for the spread even and prorata methods.

Common dimension jobs can be divided into the following five categories:

- Business_Unit
- Calendar
- Currency
- Language
- Unit_Of_Measure

The common dimension master sequence jobs can be found in the following DataStage Director paths:

- Common_Dimensions\E\Business_Unit\Master_Sequence
- Common_Dimensions\E\Calendar\Master_Sequence
- Common_Dimensions\E\Currency\Master_Sequence
- Common_Dimensions\E\Language\Master_Sequence
- Common_Dimensions\E\Unit_Of_Measure\Master_Sequence

Note. For all dimension load jobs (common dimension, global dimension, local dimension, OWE dimension, MDW dimension), users can customize the error validation by providing the environmental variable with the appropriate values. If you want to skip error validation, set $ERR_VALIDATE to 'N.' If you want to perform error validation, set $ERR_VALIDATE to 'Y.' Also, you can specify the threshold limit for the error validation. If you want the job to abort if the lookup fails more than 50 times, set $ERR_VALIDATE to 'Y' and $ERR_THRESHOLD to 50. This can all be done using DataStage Administrator.

Running Common Dimensions Jobs

Perform the following steps to run the common dimension jobs (the order reflects the master sequence order):

1. In DataStage Director, navigate to the MSEQ_E_Hash_Calendar (Calendar) master sequence by expanding the nodes in the left navigation panel using the path defined in the previous section.
2. Select the MSEQ_E_Hash_Calendar master sequence job in the Job Status view and select *Job, Run Now...* from the menu.

   The Job Run Options box appears.
3. Update the job parameters if necessary and click Run.

   The job is scheduled to run with the current date and time, and the job's status is updated to *Running.*
4. Repeat steps one through three for the remaining master sequence jobs, using the following order:
   a. MSEQ_E_OWE_BaseDim_Calendar (Calendar)
   b. MSEQ_E_OWS_BaseDim_Calendar (Calendar)
   c. MSEQ_E_Hash_BU (Business Unit)
   d. MSEQ_E_OWE_BaseDim_BU (Business Unit)
   e. MSEQ_E_OWS_BaseDim_BU (Business Unit)
   f. MSEQ_E_Hash_Currency (Currency)
   g. MSEQ_E_OWE_BaseDim_Currency (Currency)

Chapter 14

Importing Source Business Units into EPM to Create Warehouse Business Units

This chapter provides an overview of warehouse business units, tableset sharing, setID mapping, and warehouse business unit setup, and discusses how to:

- Define setID and warehouse lineage.
- Establish warehouse business units using the Business Unit Creation Wizard.
- Establish warehouse business units manually.
- Review warehouse and general ledger business unit creation.
- Define collision maps.

Understanding Warehouse Business Units, TableSet Sharing, and SetID Mapping

As a part of EPM setup, you must create *warehouse business units* (WBU), and establish a mapping between warehouse business units and the business units that exist in the source systems that you are bringing into EPM. The mapping between source business units and warehouse business units has implications for the appropriate setID assignments for warehouse business units on warehouse *record groups.* Setting up warehouse business units is mandatory regardless of whether you plan to implement only data marts, only Analytical Applications, or some combination of the two.

SetIDs and TableSet Sharing

In PeopleSoft source systems (such as Campus Solutions or CRM), the rows in a control table (such as supplier, customer, and account) are divided into groups based on a key called the set ID. Each group of rows with the same setID constitutes a *tableset.* For example the figure below shows the tablesets on a hypothetical supplier table with two setIDs represented in red and blue:

Hypothetical Supplier Table

SETID	SUPPLIER_NAME	
RED	Pepsi Inc.	
RED	Coca-cola Corp.	"RED" TableSet
RED	Seven Up Bottling	
BLUE	Frank's Bottling Co.	
BLUE	Jane's Beverages, Inc.	"BLUE" TableSet
BLUE	Seven Up Bottling	
Etc.	Etc.	

Tablesets on a hypothetical supplier table

Every business unit associates with a particular setID on every control table. For example, BU1 might use the red tableset on the supplier table. This assignment serves to limit the suppliers that appear in prompts for transactions that take place in the context of BU1. Several other business units could also use the red setID on the supplier table, in which case these business units share the red tableset (hence the term *tableset sharing*). Since control tables in a PeopleSoft source system generally become dimensions in EPM, most EPM dimensions are setID-based.

Record Groups

Control tables such as supplier are themselves grouped into record groups based on the commonality of business process. SetID assignments actually take place at the record group level. In other words, a business unit must have the same setID on all control tables in a given record group. Each business unit in a PeopleSoft source system must have a setID assignment on every record group. At the time a business unit is created, the system assigns a default setID to all record groups for that business unit. The user can then manually modify the setID assignments as desired. For example, the following table illustrates the setID assignment for a few hypothetical business units on a few hypothetical record groups:

A Hypothetical Set Control "Space"

	RG1	RG2	RG3	RG4	Etc.
BU1	RED	RED	BLUE	YEL	
BU2	RED	RED	BLUE	YEL	
BU3	RED	BLUE	BLUE	YEL	
BU4	BLUE	RED	BLUE	YEL	
BU5	BLUE	RED	RED	YEL	
BU6	BLUE	RED	BLUE	PINK	
Etc.					

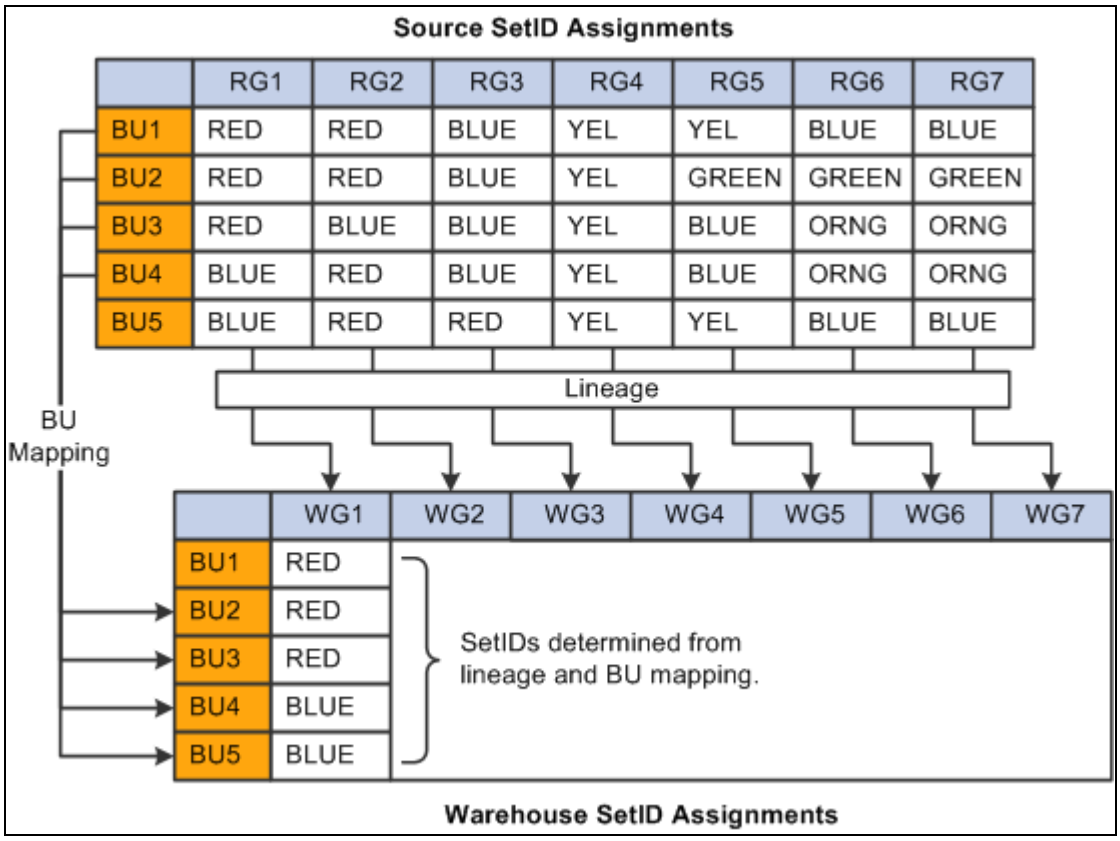
SetID assignments for hypothetical record groups

Assuming that the supplier table shown earlier is included in record group RG1, then (given the setID assignments indicated in the figure above) BU1, BU2, and BU3 would have access to the first three suppliers, while BU4, BU5, and BU6 would have access to the last three.

Mapping Two SetID Based Systems Together and Warehouse Lineage

EPM is a setID-based database. Consequently, all warehouse business units must have setID assignments on all warehouse record groups. However, because each PeopleSoft source system has its own tableset sharing (or *set control space*), the mapping between source business units and warehouse business units has implications for how setIDs must relate between the source system and the warehouse. As illustrated below, in order to correspond two set control spaces, it is necessary to provide two distinct mappings: the business unit mapping (which is under user control) and the warehouse *lineage*. Warehouse lineage defines the correspondence between all PeopleSoft source tables and EPM warehouse tables.

See Chapter 14, "Importing Source Business Units into EPM to Create Warehouse Business Units," Warehouse Lineage and Source Blueprints, page 287.

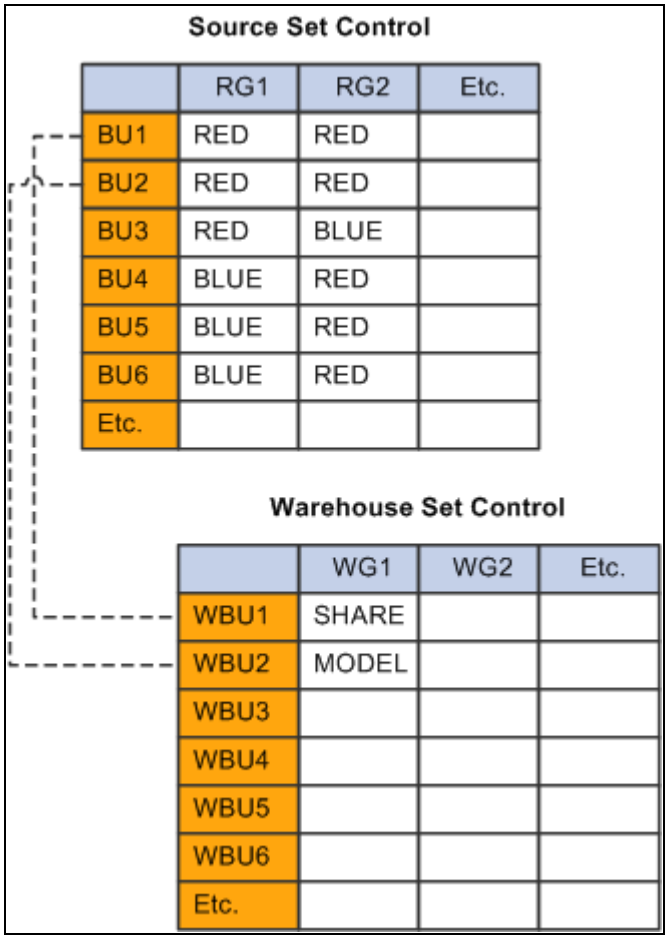


Hypothetical source to warehouse set control space mapping

The preceding figure illustrates two set control spaces and the mappings between them, with setID values shown as colored squares. In this example, warehouse business units have been created and mapped to source business units in a one-to-one relationship. Based on the lineage, it is possible to determine which setID values to assign to the corresponding warehouse record groups. Without the lineage information this determination would not be possible.

Potential SetID Conflicts

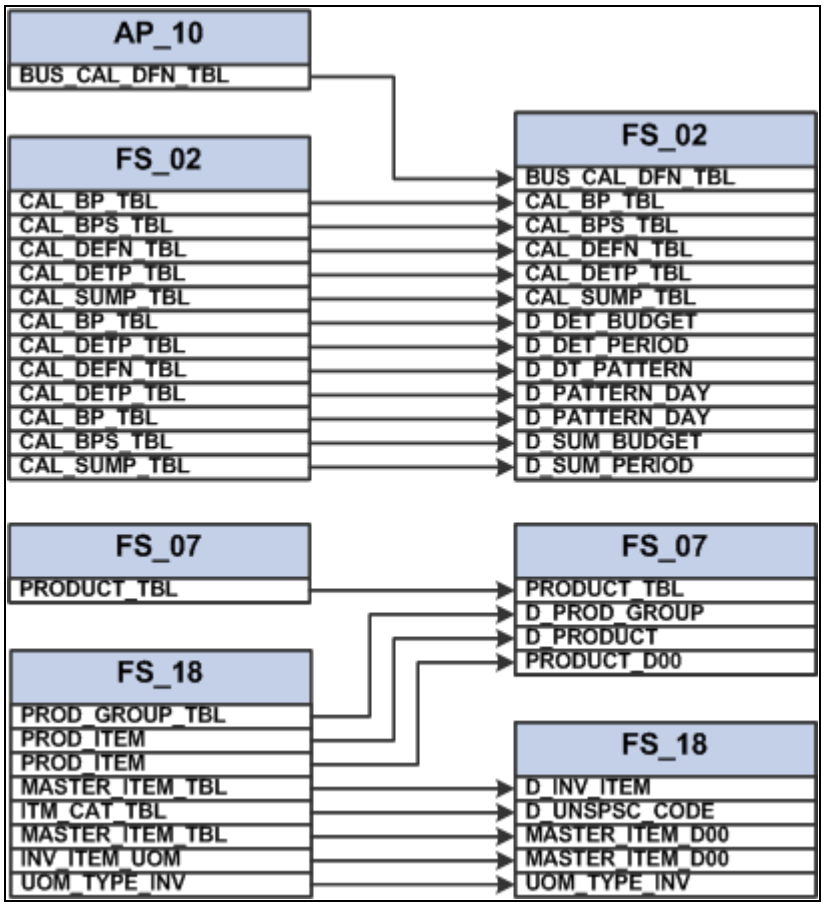
SetIDs from one or more sources can merge into a single warehouse setID (in other words, "many-to-one" relationships are allowed). However, the system does not allow one-to-many relationships to exist between source setIDs and warehouse setIDs on the same warehouse record group. For example, in the illustration below, BU1 and BU2 shared setID RED in the source. Assuming that RG1 is related by lineage to WG1, then the situation illustrated would force one setID (RED) to map to two different values (SHARE and MODEL) on warehouse record group WG1. This situation is not allowed because it would create a multi-valued lookup that would cause errors in the ETL process. Consequently the system does not allow this configuration to pass its validation step. You must modify the warehouse setID assignments to remove the splitting. In the case illustrated below, simply assigning the same warehouse setID to WBU1 and WBU2 on record group WG1 resolves the problem.



SetID splitting

FSCM and SetID Conflicts

Business unit merging is not the only way that setID conflicts can arise. There are two instances in the FSCM product that has a lineage that can introduce conflicts, as shown below:



SetID conflicts in the FSCM to warehouse mapping

The problem occurs when record groups from a source merge in the warehouses (for example, AP_10 and FS_02 merging into FS_02, as shown in the diagram above). When you use the Business Unit Creation Wizard, it assigns an initial guess for the setIDs of the warehouse business units that it creates for you. A problem can occur in the case where two or more business units have different setIDs on AP_10 and FS_02. For example, if you had two business units in FSCM (BU1 and BU2), and assigned them both setID RED on AP_10, and BLUE on FS_02, it is possible that the wizard could select RED as the setID assignment for FS_02 for WBU1, and BLUE for the setID of WBU2. In this case the wizard would detect splitting because it would see both RED and BLUE from the source being split to RED and BLUE on the target. The solution to this problem is straightforward: you should manually create the warehouse business units, and assign the correct setID. Then you can map the business units and do the validation in the Business Unit Mapping page. If you have an FSCM source with a complex setID configuration and you detect conflicts related to AP_10, FS_02, FS_07, or FS_18, it may be due to this lineage problem.

Warehouse Lineage and Source Blueprints

Warehouse lineage defines the relationship between PeopleSoft Enterprise source tables and EPM target warehouse tables, as determined by the ETL process. Certain ETL jobs move data from PeopleSoft source tables to EPM Operational Warehouse - Staging (OWS), Operational Warehouse - Enriched (OWE), and Multidimensional Warehouse (MDW) target tables, and warehouse lineage simply refers to the path the data takes between the two systems. Lineage information is used during the warehouse business unit creation process to determine the appropriate PeopleSoft source setID for a given warehouse business unit, and map that setID to the appropriate warehouse record group.

Source blueprints actually *record* the warehouse lineage, and ship with EPM as system data.

Warehouse lineage information is only required for setID-based source systems. As such PeopleSoft delivers blueprints for PeopleSoft source systems only. Typically, third-party source systems do not use a setID column, and thus have no impact on the blueprints. You need not create blueprints for non setID-based third-party source systems.

Modifying a Source Blueprint Due to Customizations

If you customize your ETL jobs or add columns to an existing source or target table , you may need to modify the source blueprints to reflect these changes. It is only necessary to update the blueprints if you add a new setID-based table to your PeopleSoft source system and map the table into new, setID-based dimension in the warehouse. In that case, simply access the Source Blueprint page and add a row for each new dimension.

Remapping Business Units for an Updated Source Blueprint

If you receive an updated Source Blueprint from PeopleSoft, you must remap business units manually or with the business unit wizard after running the dms/dat. This creates the source to target mapping. All entries must be truncated and rerun.

Understanding Warehouse Business Unit Setup

Warehouse business units are created using several processes, including ETL jobs, defining parameters in PeopleSoft pages, and running a setup and validation wizard.

The following steps are required to establish warehouse business units:

1. Run prerequisite setup jobs.
2. Create setIDs in the warehouse.
3. Define source systems.
4. Establish warehouse lineage.
5. Create warehouse business units.
6. Assign warehouse setIDs to warehouse business units on all warehouse record groups.
7. Map source business units to warehouse business units.
8. Validate the business unit mapping.

PeopleSoft delivers the *Business Unit Creation Wizard* that automates many of these steps. However, you can manually perform these steps if you prefer.

Establishing Default SetIDs - Overview

You must define a default setID for each source that you wish to bring into the warehouse. SetIDs are created in a PeopleTools page, where you can add a new value for each default setID that you need to create. Generally you should use a different default setID for each source to keep tablesets from all sources separate—unless you have a good reason why data from different tablesets should merge. The wizard uses the default setID for all business units that it creates on all warehouse record groups for which the source blueprint supplies no lineage information. For example, if you bring in a CRM source, the CRM blueprint has no lineage information for many warehouse tables that relate to HCM or FSCM, simply because there are no source tables in CRM that relate to these tables. All unspecified warehouse tables receive the default setID.

Establishing Warehouse Sources - Overview

For each PeopleSoft source that you are bringing into the warehouse, use the Define Warehouse Sources page to define the properties of your source. The Business Unit Creation Wizard uses the defaults specified on the page for warehouse business unit creation. You should define a warehouse source even for third-party systems that are not setID-based.

The procedure for defining warehouse sources will not be discussed in this chapter because it is discussed in another chapter of this PeopleBook.

See Chapter 4, "Setting Up EPM Business Rules," Defining Warehouse Sources, page 51.

Note. To review the sources you currently have defined for your system, run the PS Query *DMRP_2_1_SRC_SYSTEM_TBL*.

Establishing Warehouse Lineage - Overview

Establishing lineage consists of two parts:

- Reviewing or updating source blueprints.
- Defining warehouse lineage.

Review or Update Source Blueprints

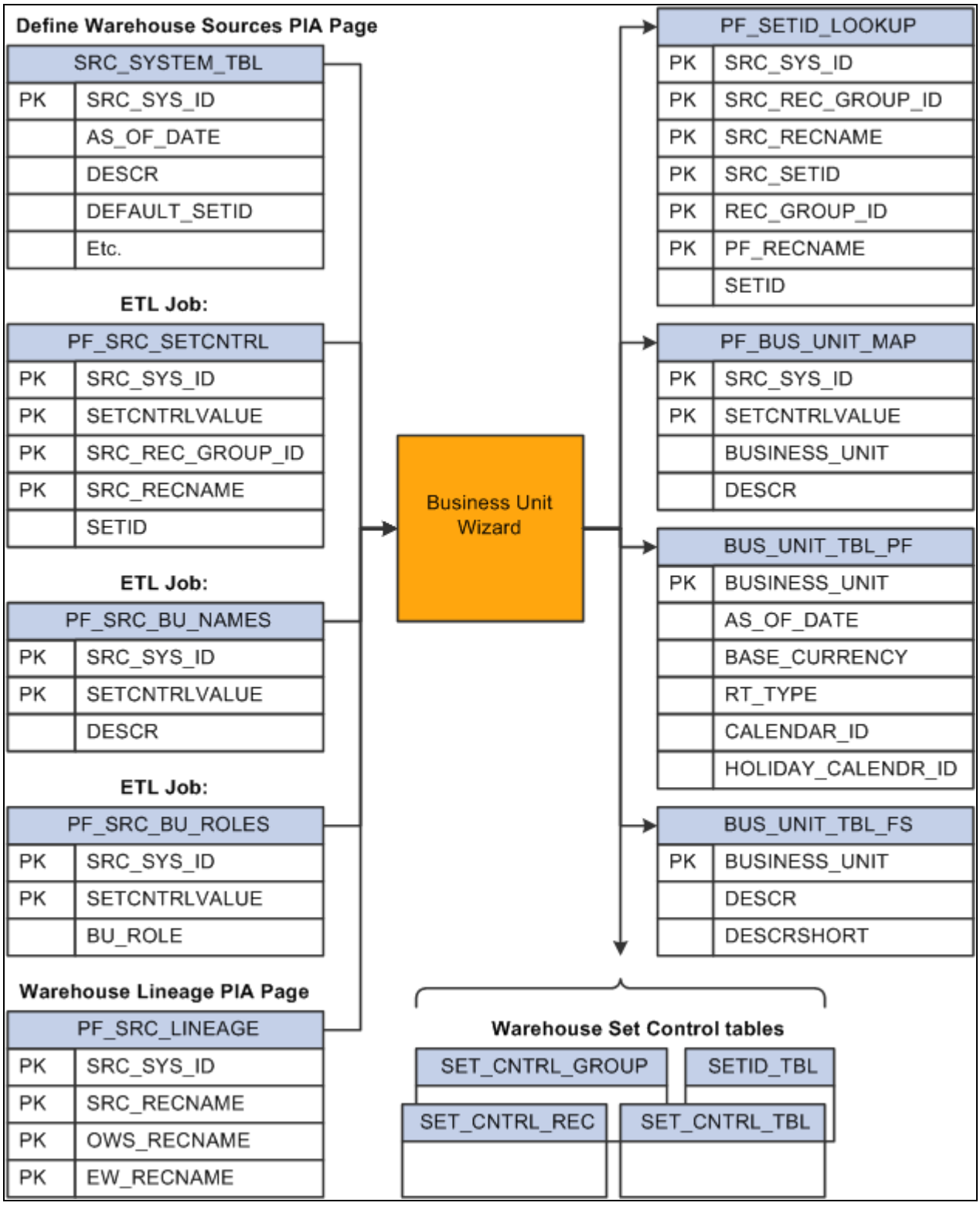
Source blueprints are used in establishing warehouse lineage. PeopleSoft delivers source blueprints for PeopleSoft sources. If you have added new SetID-based tables to your PeopleSoft system and you wish to bring these tables into the EPM database, you must update the source blueprints accordingly.

Define Warehouse Lineage

After you have defined your sources, you must associate a blueprint to each source by using the Warehouse Lineage page. The system ships with blueprints for all supported systems. Note that the FSCM Blueprint includes lineage information for ESA, SCM, and PeopleSoft Financials. Enterprise Learning Management (ELM) is not setID-based, so you can use the NONSETID blueprint for ELM sources (you should also use the NONSETID blueprint for third-party systems that have no setID). When you save the warehouse lineage page the system populates the PF_SRC_LINEAGE table with the combined lineage for all warehouse sources.

Creating Warehouse Business Units with the Business Unit Creation Wizard - Overview

The Business Unit Creation Wizard automates many of the steps required to set up warehouse business units and setIDs. Before you can use the wizard, you should understand the input and output tables used by the wizard. The following diagram illustrates the inputs that the wizard uses, and the output tables that it populates:



Input and output tables used by the Business Unit Creation Wizard

The Business Unit Creation Wizard uses the following input tables:

- SRC_SYSTEM_TBL: This table defines all the pillars that you are bringing into EPM. You populate this table using a PIA page

- PF_SRC_SETCNTRL: This table records the set controls (a combination of business units and setIDs) from all of your sources, and the setID assignments for all set controls on all source record groups. It is populated by ETL.
- PF_SRC_BU_NAMES: This table provides a distinct list of business units from all sources. It is populated by ETL
- PF_SRC_BU_ROLES: This table records all of the roles that associate to each source business unit, such as AP (accounts payable business unit), GL (general ledger business unit), and IN (inventory business unit).
- PF_SRC_LINEAGE: This table is a compilation of all of the Source Blueprints that you specify for all active sources that you create. It is populated by a PIA page when you save the Blueprint assignments (see below).

The Business Unit Creation Wizard populates the following output tables:

- PF_SETID_LOOKUP: The ETL process uses this table to map source setIDs to warehouse equivalents.
- PF_BUS_UNIT_MAP: The ETL process uses this table to map source business units to warehouse equivalents.
- BUS_TABLE_PF: The Wizard makes entries in this table for all warehouse business units that it creates. These tables are used by the analytical applications.
- BUS_TABLE_FS: Same as BUS_UNIT_TABLE_PF
- Warehouse Set Control Tables: The wizard makes the necessary entries in the warehouse set control tables when it assigns setIDs to each warehouse business unit on each warehouse record group.

The Business Unit Creation Wizard consists of three steps:

1. Reviewing Incoming Set Controls, Business Unit Wizard Step 1: Review the set controls from all of your sources and ensure you see the business units and setIDs that you expect from each source.
2. Addressing Collisions Between Set Controls, Business Unit Wizard Step 2: Examine name collisions between and among incoming set controls and set controls that may already exist in the warehouse. You should resolve the name conflicts by renaming the incoming set control unless you have a good reason to merge them.
3. Validating the Mapping, Business Unit Wizard Step 3: After the wizard has analyzed the setID assignments for all incoming set controls, you must review the mapping report produced by the wizard to see if any conflicts exist. If conflicts exist, you must correct the conflicts yourself or let the wizard correct the conflicts.

Note. Prior to running the Business Unit Creation Wizard, you must run the ETL jobs that populate the aforementioned input tables. These jobs are collectively described as dimension mapper ETL setup jobs. The Wizard populates the tables indicated above as output tables. For a new EPM installation, the output tables are empty. In the event that you make a mistake and need to re-run the wizard, you must truncate the output tables manually; no provision is made for the wizard to delete these entries. If you have already loaded data into the warehouse and then decide to re-run the wizard, the data you have already loaded may become invalid.

Creating Warehouse Business Units Manually - Overview

You can manually setup warehouse business units. The steps are outlined here:

Create Warehouse Business Units

Manually define all the warehouse business units that you require, using the Warehouse Business Unit page. You must define a default setID, a default currency rate type, and a calendar for each warehouse business unit.

Assign Warehouse SetIDs to Warehouse Business Units

As you create warehouse business units, you are prompted to provide a default setID. The default setID is automatically assigned to the new business unit on all record groups. In order to modify these assignments, you must use the Tableset Control page. Select the business unit (set control) of interest and modify its setID assignments on the record groups of your choice.

If several warehouse business units have exactly the same setID assignments, it's possible to copy the setIDs of an existing business unit. Define the setID assignments for one warehouse business unit, and then as you create new ones, use the name of the first warehouse business unit as the default setID of subsequent warehouse business units. The system assigns each record group the same setID as the copied business unit.

Map Warehouse Business Units

Once warehouse business units have been created, you must map them to source business units using the Business Unit Mapping page. It's possible to associate several source business units with a single warehouse business unit, thus creating a many-to-one mapping. However this is not good practice since it promotes setID conflicts, and it could cause fact table collisions as several business keys merge into one. PeopleSoft delivered ETL jobs do not support aggregating fact data during loading.

Validate Business Unit Mapping

Once you have created your mapping configuration, validate the mapping. The system analyzes the mapping configuration. If all of the set controls have been made unique as suggested above, the potential for conflicts in your business unit mapping is minimized. If conflicts are detected, you have two choices:

- Let the system correct the conflicts: The simplest choice is to click the "Accept Proposed setIDs and Save" button. This causes the system to automatically reassign the setIDs used by warehouse business units on various warehouse record groups in order to make the configuration valid. Generally this reduces the number of warehouse setIDs to eliminate conflicts.
- Modify business unit Mapping: You can choose to resolve the conflicts manually. This requires that you analyze the report to determine which warehouse setIDs must change in order to avoid setID Splitting.

Regardless of how you fix the problem, if conflicts are detected, you must return to the business unit mapping page and try the validation again, until no conflicts exist. Continue to iterate the validation process until you see no more conflicts. Once the conflicts are resolved, the process of creating business units and assigning setIDs is complete. You only need to inspect and modify the properties of the new warehouse business units.

Working with Invalid or Unused Source Business Units - Overview

The ETL logic that extracts source business unit data and brings it into the EPM database does not filter-out invalid or unused source business units. Hence, if you have invalid or unused business units in your PeopleSoft source, you can:

- Create a dummy Warehouse Business Unit (WBU) and map the undesired source business units to the dummy WBU.
- Reconfigure the Dimension Mapper setup jobs to filter out the source business units that you do not want to bring into the EPM database.

However, it is not recommended that you reconfigure the Dimension Mapper setup jobs or use SQL to delete business units from the internal Dimension Mapper tables (the unwanted business units will reappear the next time you run the Dimension Mapper setup jobs).

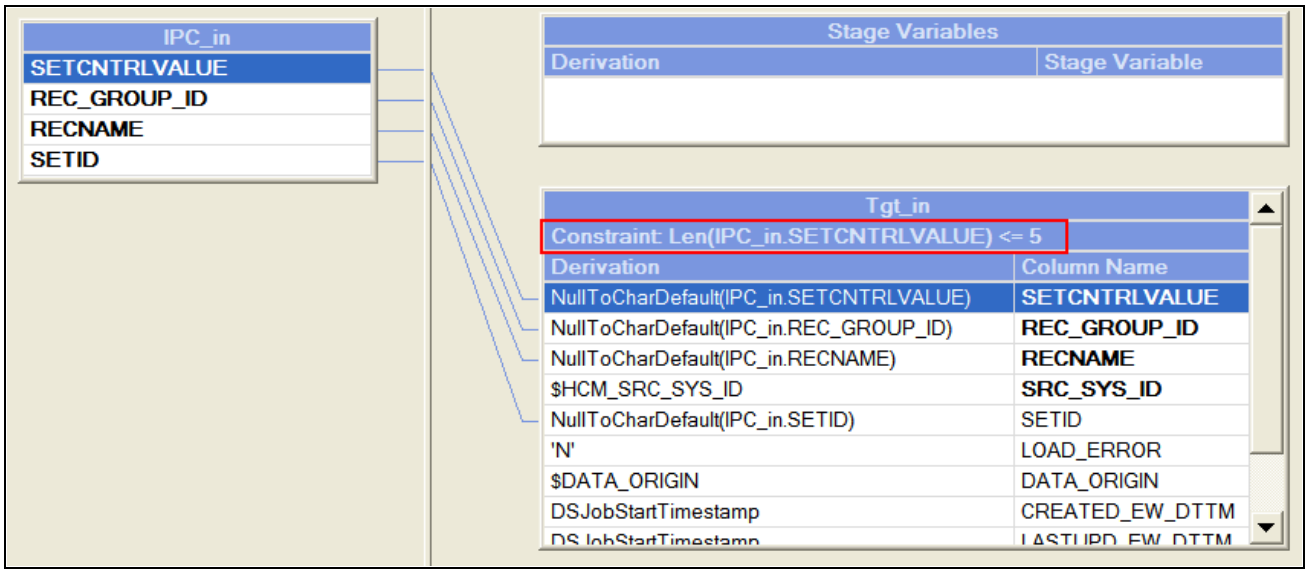
Working with PeopleSoft Enterprise HRMS Source Business Units - Overview

As part of your configuration of the PeopleSoft Enterprise HRMS source system, you used the Company component to enter information about a single company or multiple companies in your organization. You assigned a three-character code for each company you defined in the HRMS source system. During the ETL process, those company codes are brought into the PF_SRC_SETCNTRL table, which records the source set controls, and is used as an input for the warehouse business unit mapping process.

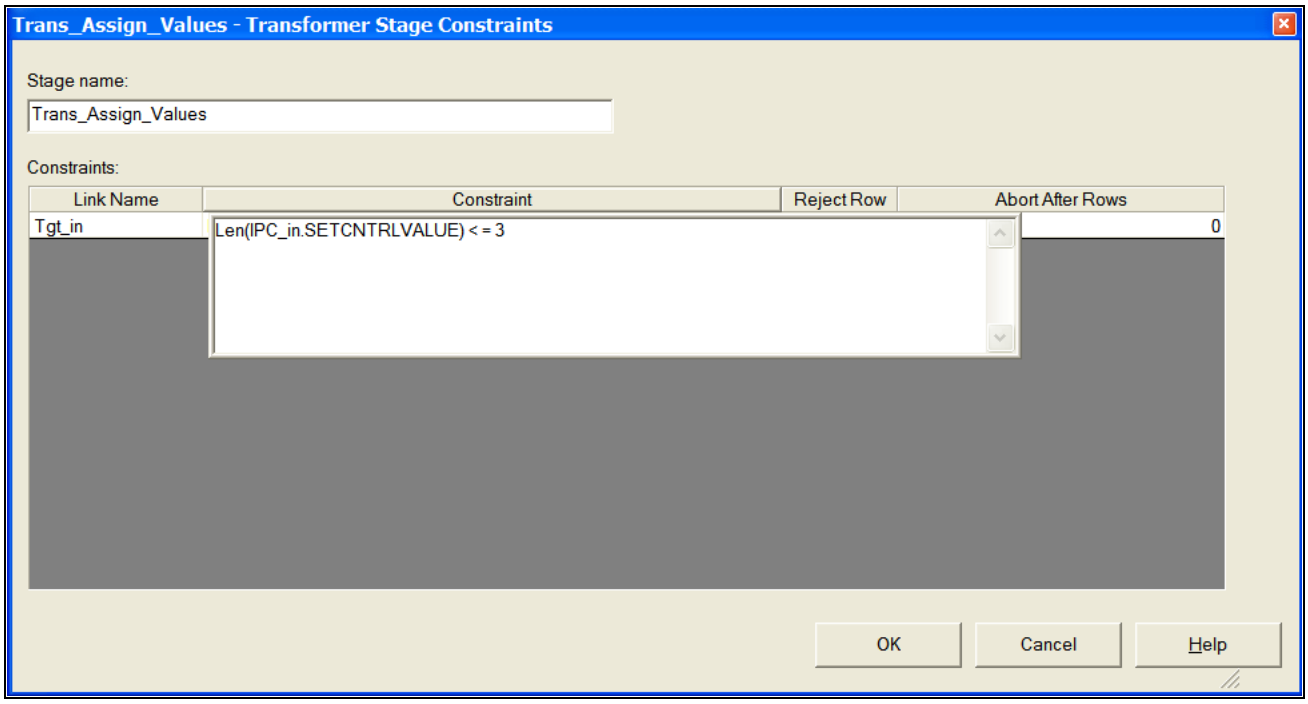
The company codes reside in the same table (PF_SRC_SETCNTRL) as your source business units. Hence, when you map source business units to warehouse business units, be sure not to include the company codes in your mapping. If you have a large number of company codes, you may wish to filter-out these codes in the ETL job.

Filtering Company Codes in the ETL Job - Trans_Assign_Values Stage

If you are sure that all SETID and BUSINESS_UNIT values are five-characters in length (as is recommended), and all company codes are three-characters in length, you can access the ETL job *J_Stage_PS_S_SET_CNTRL_REC_HCM_HCM91_EPM91* using DataStage Designer and filter all rows with Len(IPC_in.SETCNTRLVALUE) < = 3 in the Trans_Assign_Values Stage.



Modifying the Stage Constraint, 1 of 2

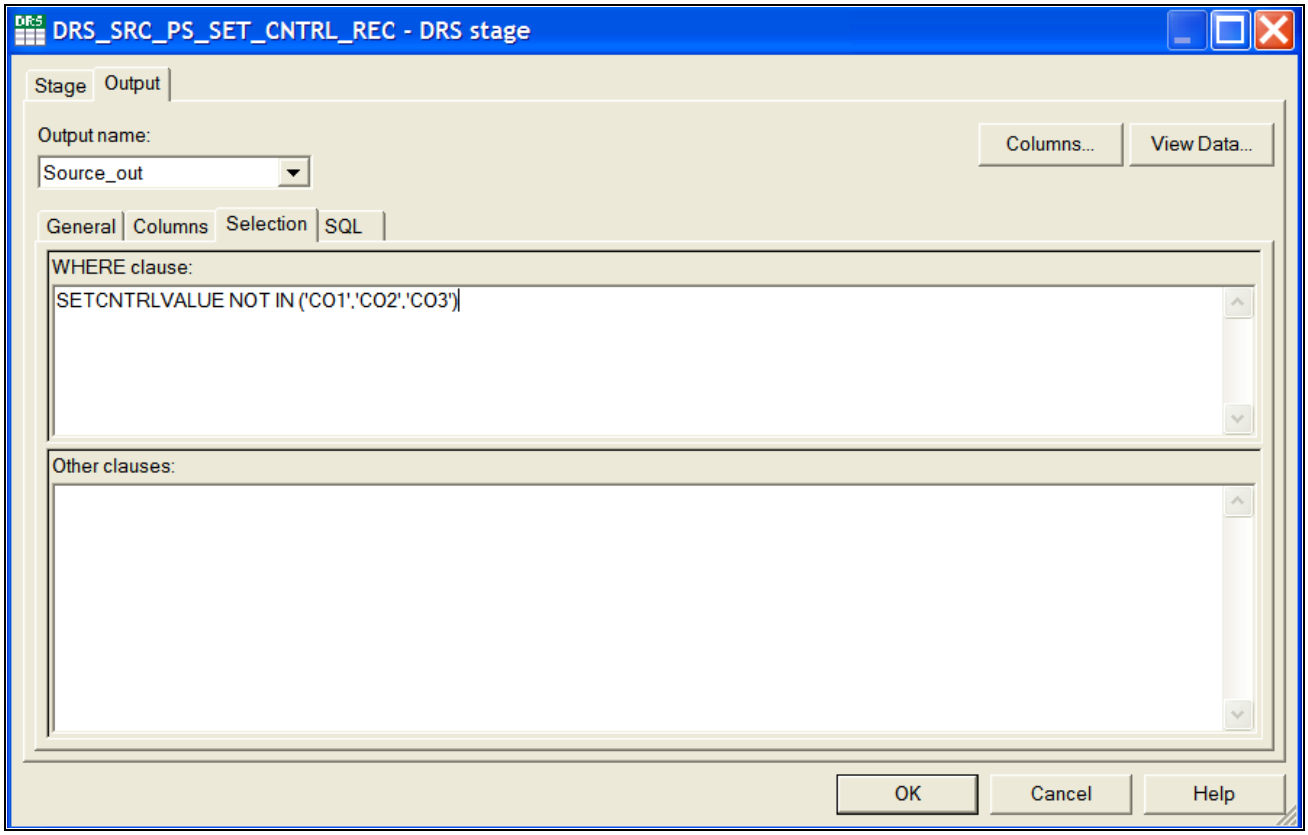


Modifying the Stage Constraint, 2 of 2

However, if you perform this configuration and you set up table-set sharing using company as the driving parameter along with business unit, the same table-set sharing information would be prevented from entering EPM and the business unit wizard would *not* be able to retain table-set sharing based on company codes in the HRMS source system.

Filtering Company Codes in the ETL Job - Source Stage

Alternatively you can create a WHERE filter in the selection-output of the source stage (DRS_SRC_PS_SET_CNTRL_REC) in the ETL job *J_Stage_PS_S_SET_CNTRL_REC_HCM_HCM91_EPM91*.



Creating a WHERE filter in the selection-output of the source stage

The WHERE filter above is just an example, it is not meant to be the exact WHERE filter you should create for the source stage.

Reviewing Warehouse and General Ledger Business Unit Properties - Overview

After creating and mapping warehouse business units, you should review your warehouse business units and general ledger business units to ensure the properties (such as default calendar) meet your requirements.

Review Warehouse Business Unit Properties (Business Unit Creation Wizard Only)

After creating warehouse business units with the Business Unit Creation Wizard, access the Warehouse Business Unit page to review the detailed properties of each business unit. Some of your warehouse business units may have been created with a base currency or rate type that differs from the defaults that you defined for your source. If this is the case, you need to change these settings for the appropriate business units. In addition, the Business Unit Wizard does not associate calendars to business units. You must do this manually for all your warehouse business units.

Review General Ledger Business Unit Properties

PeopleSoft general ledger business units (GLBU) are extracted from your source system and populated in the EPM database using ETL jobs; you do not need to recreate them in the warehouse. You can view general ledger business units by accessing the General Ledger Business Unit page. You may, on some occasions, create general ledger business units manually in the warehouse for certain analytical applications. See your EPM application PeopleBook for more details.

Creating Collision Maps - Overview

If you choose to allow two tablesets to merge in the EPM database, it is possible for collisions to occur between business keys (such as supplier ID). For example two suppliers from two different tablesets could both have the business ID "PEP", but could refer to very different suppliers (for example, Pepsi and Pep Boys). Collision maps provide a framework for resolving collisions between business IDs from two or more tablesets that merge in the warehouse. PeopleSoft provides three resolution methods for colliding business IDs: First In Wins, Error-Out Duplicates, and Use Mapping Table.

First In Wins

In this method, the first instance of a business ID is loaded into a tableset, and subsequent instances of the same ID from different tablesets is ignored. This approach is appropriate for large datasets, when the number of collisions is known to be small and the value of fixing errors is low. You can inspect the business keys that have been ignored by navigating to the Collision Map Error Report page.

Error Out Duplicates

In this method the system automatically loads the first instance of a business ID into a tableset, but subsequent instances of the same ID from different tablesets is sent to an error table where they can be inspected and remapped. This approach is appropriate for relatively small datasets, where the value of fixing errors is high. You must inspect and correct the errors manually by navigating to the Collision Map Error Reports page for the appropriate map.

Use Mapping Table

When using this mapping type, the system checks every incoming business key against a mapping table. If an entry does not exist in the mapping table, then the row errors-out. You can inspect the errors in the appropriate error report. For those entries that error out, you must update the mapping table and re-run the appropriate ETL map. This mapping method is relevant to the case where you have created a mapping table using an offline process or third-party tool.

Prerequisites to Creating Warehouse Business Units

This section provides you with prerequisites to creating warehouse business units and discusses how to:

- Run prerequisite ETL setup jobs.
- Size tablespaces for input tables.
- Create backups of impacted output tables.

- Enable PS Queries
- Verify the state of output tables
- Validate business unit and set control data
- Verify source blueprints

Running Prerequisite ETL Setup Jobs

Before you can begin creating warehouse business units you must run specific ETL jobs that setup certain OWS and OWE tables, and bring your source business unit data into EPM tables. Some of these jobs include OWS hash file, setup OWS, shared lookup, and setup OWE jobs.

The general OWS and OWE ETL setup jobs are not discussed here because the information is covered in another chapter of this PeopleBook. However, the ETL jobs that are specific to business unit data are discussed below.

See Chapter 11, "Setting Up DataStage for EPM," page 255 and Chapter 13, "Running Initial Setup Jobs," page 275.

Running Hash File - Business Unit Data Jobs

To run the hash file jobs that pertain to business unit data:

- In DataStage Director, navigate to the hash file jobs for business unit data using the following navigation: *Setup_E, Dimension_Mapper, Base, Load_Hash_Files, Server.*
- Select the jobs in this category and run.

Running Setup - Business Unit Tables Jobs

To run the setup jobs that load business unit setup tables:

- In DataStage Director, navigate to the hash file jobs for business unit data using the following navigation: *Setup_E, Dimension_Mapper, Base, Load_Tables, Sequence.*
- Select the jobs in this category and run.

Running Shared Lookup - Business Unit Data Jobs

To run the shared lookup jobs that pertain to business unit data:

- In DataStage Director, navigate to the shared lookup jobs for business unit data using the following navigation: *Shared_Lookups, DimensionMapper_Lookups*
- Select the jobs in this category and run.

Sizing Tablespaces for the Input Tables

The Business Unit Creation Wizard input tables use the *EWLARGE* tablespace , while the *PSINDEX* tablespace is used for indexes. These table spaces should be large enough to store the data in input, output and intermediate tables related to warehouse business unit creation.

A good rule of thumb is to make sure that each of the two tablespaces are at least twice the size of the PF_SRC_SETCNTRL table. Determine the maximum size for a row of data in PF_SRC_SETCNTRL by examining this table in your database. Estimate the number of rows expected in this table, and then multiply these factors together. Double the result to get a minimum tablespace size estimate: Minimum tablespace size = 2 * #Rows * Size of one row of PF_SRC_SETCNTRL.

Creating Backups of Impacted Output Tables (Optional)

Create a DAT file backup of the output tables that are impacted by the warehouse business unit creation process. That way, if you encounter problems during the creation process you can use the DAT file to roll back your system.

Run the prepackaged Data Mover script *DMBK_CREATE_DM_BACKUP* to produce the backup DAT file of the impacted output tables. The following are the output tables backed up by the DAT file:

- SETID_TBL
- SET_CNTRL_TBL
- SET_CNTRL_GROUP
- SET_CNTRL_REC
- BUS_UNIT_TBL_FS
- BUS_UNIT_TBL_PF
- BUS_UNIT_TBL_GL
- PF_BUS_UNIT_MAP
- PF_SETID_LOOKUP

To roll back your system, run the Data Mover script *DMBK_RESTORE_TABLES.*

Enabling PS Queries for Data Verification

The following sections provide instructions on how to verify different aspects of your business unit related data. To perform the verifications you must run certain prepackaged PS Queries. To obtain the specific queries you must first enable the queries and the query security by installing the *DMRP_QUERIES* Application Designer project.

Perform the following steps to install the DMRP_QUERIES project and enable prepackaged PS Queries:

1. Locate the DMRP_QUERIES Application Designer project on My Oracle Support and install it.

2. In PIA navigate to *QUERY_TREE_EW* using the following path: *PeopleTools, Security, Query Security, Query Access Manager.*
3. Enter QUERY_TREE_EW for the tree name and search.
4. Click the QUERY_TREE_EW link from the grid to access.
5. Click the Insert Child Group button.
6. Enter *DM_RED_PAPER _GROUP* for the access group and add.
7. Enter *Dimension Mapper Red Paper* for the description and click OK.
8. Select the *DM_RED_PAPER _GROUP - Dimension Mapper Red Paper* link and click the *Insert Child Record* button.
9. Enter *BUS_UNIT_SRC_PF* for the record and add.
10. Repeat the steps to insert the following records:
    - PF_BLUEPR_DFN
    - PF_BLUEPR_DTL
    - PF_BUS_UNIT_MAP
    - PF_SCR_TBL
    - PF_SETID_LOOKUP
    - PF_SRC_BU_NAMES
    - PF_SRC_BU_ROLES
    - PF_SRC_LINEAGE
    - PF_SRC_SETCNTRL
    - PF_SSCL_DFN
    - PF_SSCL_DTL
    - SRC_SYSTEM_TBL
11. Click Save.

Once the queries are installed you can access them as follows:

1. In PIA navigate to the queries using the following path: *Reporting Tools, Query, Query Viewer.*
2. Enter *DMRP* in the search box.
3. Click on the HTML link next to the query of your choice to view the results in a browser window.

Verifying the State of Output Tables (Optional)

The prepackaged PS Queries enable you to inspect the state of the output tables that Dimension Mapper populates. If you are populating a warehouse for the first time, the output tables are empty of data except for the set control tables *MODEL* and *SHARE*, which will contain setID data.

If you are upgrading an existing warehouse, the tables are populated with data from all previously installed products. Existing warehouse business units can have an impact on the setup process if you map incoming business unites to existing warehouse business units. The Business Unit Creation Wizard respects the setID assignments of the existing warehouse business units and attempts to correlate the incoming setIDs and the existing setIDs on the appropriate record groups. If, for some reason, extraneous business units exist in the warehouses, unwanted setID mappings may be created.

For example, if you manually create a business unit named US01 in the OWE, you must assign a setID to US01 on all warehouse record groups. A default setID is required at business unit creation time, although you can manually reassign the setID for US01 on any warehouse record group by accessing the PeopleTools pages (should you wish).

If you bring in another source business unit (assume it is also named US01), and you map the two together, the system attempts to create mappings between the setIDs used by US001 in the source, and the setIDs used by US001 in the OWE. These mapping are then validated to ensure that no setID conflicts exist. If the US001 entry in the warehouse is in fact erroneous (demo data, for example), then you may create unnecessary conflicts that will impact the setup process.

Likewise, if you bring in a setID such as SHARE, and SHARE already exists in the warehouse, the Business Unit Creation Wizard prompts you to merge the two setIDs. If you select yes, then any setID assignments that SHARE itself might have in the source will be mapped to the existing setID assignments that SHARE has in the warehouse. This "second-order" indirection is utilized by some applications to enable advanced prompting features. Consequently you should make sure that you do not have erroneous WBUs or setIDs prior to running Dimension Mapper.

For these reasons, it is important to check the state of the output tables because existing business units and setIDs impact how incoming business units and setIDs from the source are mapped to their warehouse counterpart.

To inspect the state of the relevant input tables, you can use the following PS Queries:

PS Query Object	***Description***
DMRP_1_1_SET_CNTRL_REC	View the values for SET_CNTRL_REC.
DMRP_1_2_SET_CNTRL_GROUP	View the values for SET_CNTRL_GROUP.
DMRP_1_3_SET_CNTRL_TBL	View the values for SET_CNTRL_TBL.
DMRP_1_4_SETID_TBL	View the values for SETID_TBL.
DMRP_1_5_BUS_UNIT_TBL_PF	View the values for BUS_UNIT_TBL_PF.
DMRP_1_6_BUS_UNIT_TBL_FS	View the values for BUS_UNIT_TBL_FS.
DMRP_1_7_BUS_UNIT_MAP	View the values for BUS_UNIT_MAP.

PS Query Object	Description
DMRP_1_8_SETID_LOOKUP	View the values for SETID_LOOKUP.

Validating Business Unit and Set Control Data

Before you can begin creating warehouse business units or using the Business Unit Creation Wizard, you must run specific ETL job that setup the OWS, OWE, and common tables, and bring in your source business unit data. Use the PS Queries listed in the following subsections to ensure that the appropriate setup jobs have been executed, the necessary data resides in your tables, and there is no corrupt data.

Source Business Unit Data

Run the queries listed below to ensure your source business unit data is present in EPM. All business units from each source should be present. If any are missing, check the ETL jobs and run again. If extraneous business units are present (perhaps you imported the wrong data) then you will have to truncate the PF_SRC_BU_NAMES table manually.

PS Query Object	Description
DMRP_3_1_SRC_BU_NAMES	This query lists the source business units that are present in PF_SRC_BU_NAMES.
DMRP_3_2_SRC_BU_ROLES	This query lists the roles present in PF_SRC_BU_ROLES. A role is a associated with a business unit and relates to the type of business unit, such as accounts payable (AP), financial (FS), general ledger (GL), or inventory (INV). Each source BU can have one or more roles. **Note.** All business units must have the FS role.
DMRP_3_3_SETIDS	This query identifies the source setIDs. Any set control that is not present in the PF_SRC_BU_NAMES is interpreted as a setID. **Note.** If all of your setIDs are also used as business units in the source, no rows display. This does not indicate a problem.

Source Set Control Data

Run the query *DMRP_3_4_SRC_SETCNTRL* to ensure your source set control data is present in EPM. Set control information from all source systems should be present in PF_SRC_SETCNTRL and the query provides a count of the number of rows of set control information for each source that you have extracted into the system.

The number of rows returned for this query vary depending on the number of setID based and set control tables in your source. You can determine how many rows should be present in the query results by multiplying the number of setID-based tables in each source by the number of set controls that you defined in that source. The number of set controls for a given source is equal to the number of business units in that source plus the number of setIDs in that source.

For example, if you have an FSCM source with 30 business units and one setID, you should expect approximately 129,363 rows [(30 BUs + 1 setID) * (4,173) = 129,363]. If you have more than one pillar then repeat this calculation for each pillar and add up the totals. Your estimate should come very close to the results of the query, with a 10-15 percent deviation at most. If the results are significantly different, check the ETL process for errors.

The following table is provided for you to estimate the number of rows in PF_SRC_SETCNTRL.

Source System	*Number of SetID-Based Records*		*Number of Set Controls in each Source*		*Expected Number of Rows*
CRM 9.0	1,350	*		=	
ELM	98	*		=	
FSCM 9.0	4,173	*		=	
HCM 9.0	784	*		=	

Corrupt Source Data

Occasionally corrupt data can make its way into your source system (for example, someone enters data in the back-end rather than the using PIA pages, which control data quality). Corrupt data can affect your business units and set controls, and should not be allowed to enter EPM. As such, the following queries are provided to capture some business unit and set control error conditions:

PS Query Object	*Description*
DMRP_4_1_DANGLING_BU	Identifies any occurrences of business units that are not found in the Source Set Control table. **Note.** The Business Unit Creation Wizard does not create warehouse business units for source business units that do not appear in the Business Unit Names table. You must create these business units manually.
DMRP_4_2_BAD_SETCNTRLS	Identifies any business unit names that are greater than five character in length, null, or contain only a dash. These values must be removed or the validation step of the Business Unit Creation Wizard may hang indefinitely, causing the system to time-out.

Verifying Source Blueprints

Source blueprints are vital in determining warehouse lineage and are delivered as system (SYS) data in EPM. Use the *DMRP_2_2_BLUEPR_DFN* query to confirm that the blueprints are present in the EPM system and populated with the correct data.

The table, PF_BLUEPR_DFN, should contain rows for all supported source systems. You should have the following number of rows for each source system:

- CRM90: 50
- FSCM90: 174
- HCM90: 69
- NONSETID: 1

Note. These numbers may change slightly due to updated bundle fixes.

Establishing Default Set IDs, TableSets, and Warehouse Lineage

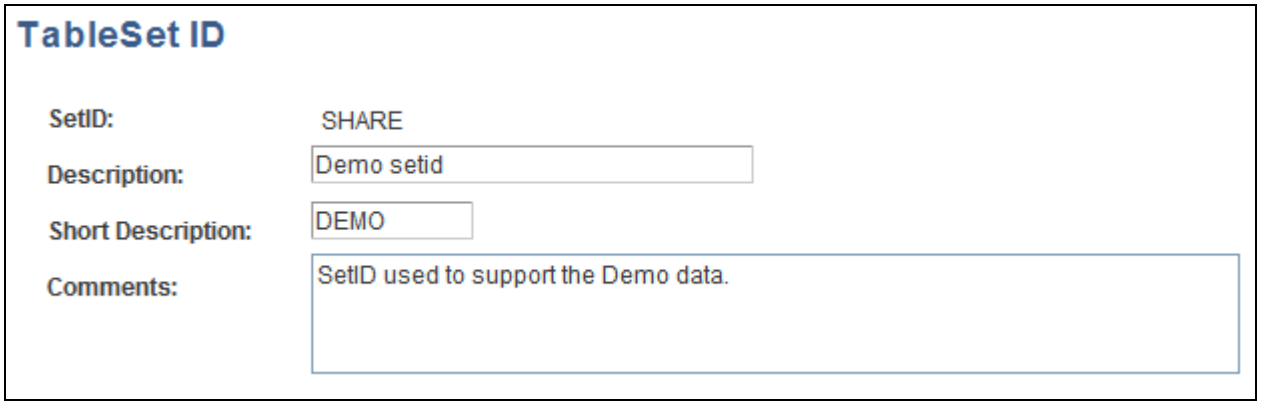
Before you can create your warehouse business units, you must define setIDs and warehouse lineage. This process identifies the source and warehouse tablesets, and the relationships (lineage) that exist between the two.

Pages Used to Define Default Set IDs, Warehouse Sources, and Lineage

Page Name	*Definition Name*	*Navigation*	*Usage*
TableSet ID	SETID_TABLE	PeopleTools, Utilities, Administration, TableSetIDs, TableSet ID	Create default setIDs for each source that you wish to bring into EPM.
Source Blueprint	PF_BLUEPRINT	EPM Foundation, EPM Setup, Warehouse Sources and Bus. Units, View Source Blueprint, Source Blueprint	Review or update selected Enterprise source blueprints.
Warehouse Lineage	PF_WHOUSE_LINEAGE	EPM Foundation, EPM Setup, Warehouse Sources and Bus. Units, Warehouse Lineage	Associate a source blueprint to each PeopleSoft source to define the lineage between source tables and warehouse tables.

Creating Default SetIDs

Access the TableSet Control page (PeopleTools, Utilities, Administration, TableSetIDs, TableSet ID).



TableSet ID page

Use this page to define a default setID for each source that you wish to bring into the warehouse. You should use a different default setID for each source to keep tablesets from all sources separate—unless you have a good reason why data from different tablesets should merge.

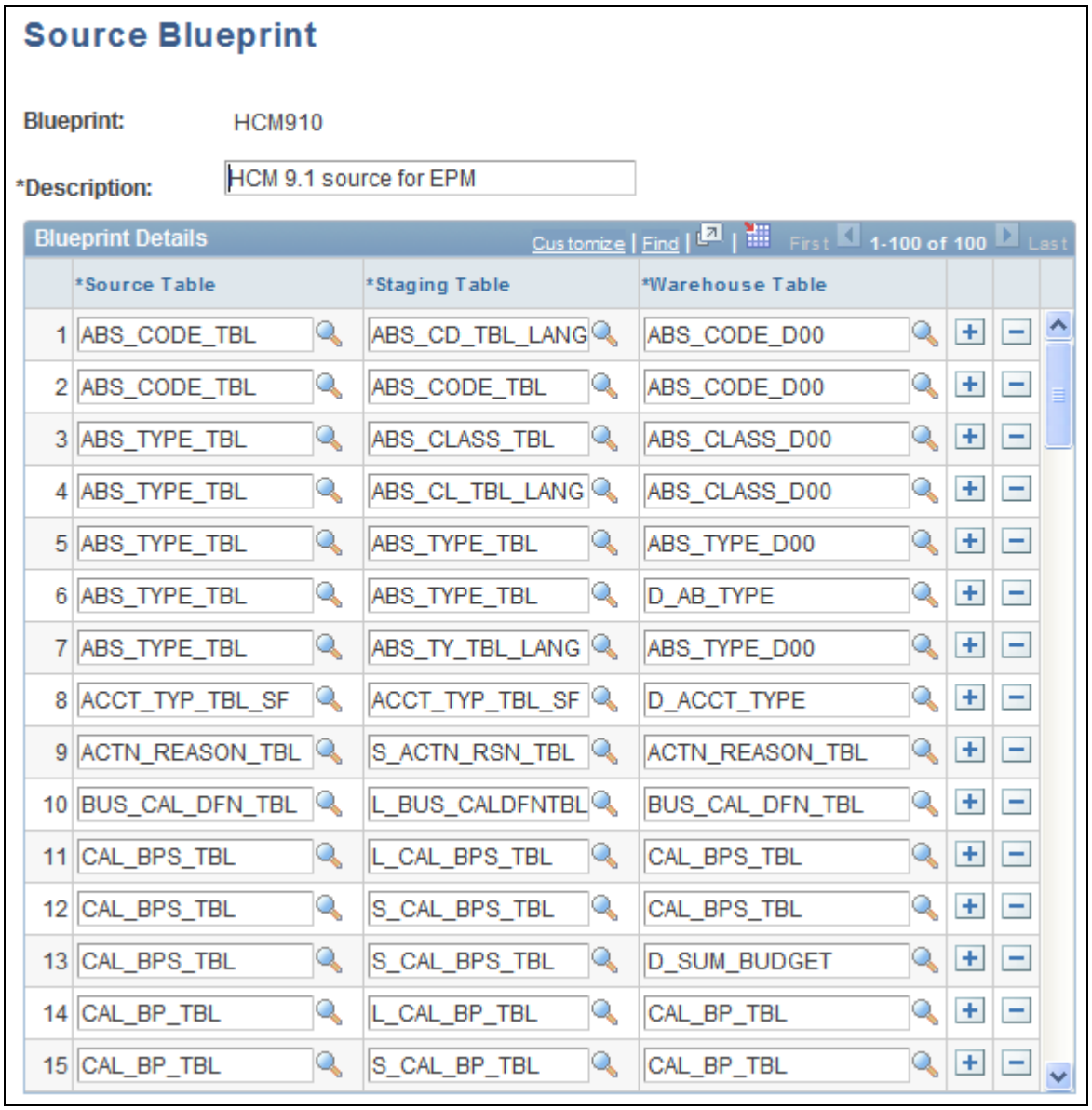
This page is discussed in detail in the PeopleTools PeopleBooks.

See *PeopleSoft Enterprise PeopleTools PeopleBook: PeopleSoft Application Designer Developers Guide*

Review or Update Source Blueprints

Access the Source Blueprint page (EPM Foundation, EPM Setup, Warehouse Sources and Bus. Units, View Source Blueprint, Source Blueprint).

Note. Before you access this page, run the ETL job that populates the PS_PF_SRC_SETCNTRL table. Otherwise, the page cannot display all the source blueprint details correctly.



Source Blueprint

Blueprint: HCM910

*Description: HCM 9.1 source for EPM

Blueprint Details Customize | Find | First 1-100 of 100 Last

	*Source Table	*Staging Table	*Warehouse Table
1	ABS_CODE_TBL	ABS_CD_TBL_LANG	ABS_CODE_D00
2	ABS_CODE_TBL	ABS_CODE_TBL	ABS_CODE_D00
3	ABS_TYPE_TBL	ABS_CLASS_TBL	ABS_CLASS_D00
4	ABS_TYPE_TBL	ABS_CL_TBL_LANG	ABS_CLASS_D00
5	ABS_TYPE_TBL	ABS_TYPE_TBL	ABS_TYPE_D00
6	ABS_TYPE_TBL	ABS_TYPE_TBL	D_AB_TYPE
7	ABS_TYPE_TBL	ABS_TY_TBL_LANG	ABS_TYPE_D00
8	ACCT_TYP_TBL_SF	ACCT_TYP_TBL_SF	D_ACCT_TYPE
9	ACTN_REASON_TBL	S_ACTN_RSN_TBL	ACTN_REASON_TBL
10	BUS_CAL_DFN_TBL	L_BUS_CALDFNTBL	BUS_CAL_DFN_TBL
11	CAL_BPS_TBL	L_CAL_BPS_TBL	CAL_BPS_TBL
12	CAL_BPS_TBL	S_CAL_BPS_TBL	CAL_BPS_TBL
13	CAL_BPS_TBL	S_CAL_BPS_TBL	D_SUM_BUDGET
14	CAL_BP_TBL	L_CAL_BP_TBL	CAL_BP_TBL
15	CAL_BP_TBL	S_CAL_BP_TBL	CAL_BP_TBL

Source Blueprint page

Source Table Displays the source table associated with the selected PeopleSoft Enterprise source.

If you have customized your ETL jobs or added rows or columns to your tables, you may need to modify this field.

Staging Table Displays the OWS staging table associated with the preceding source table.

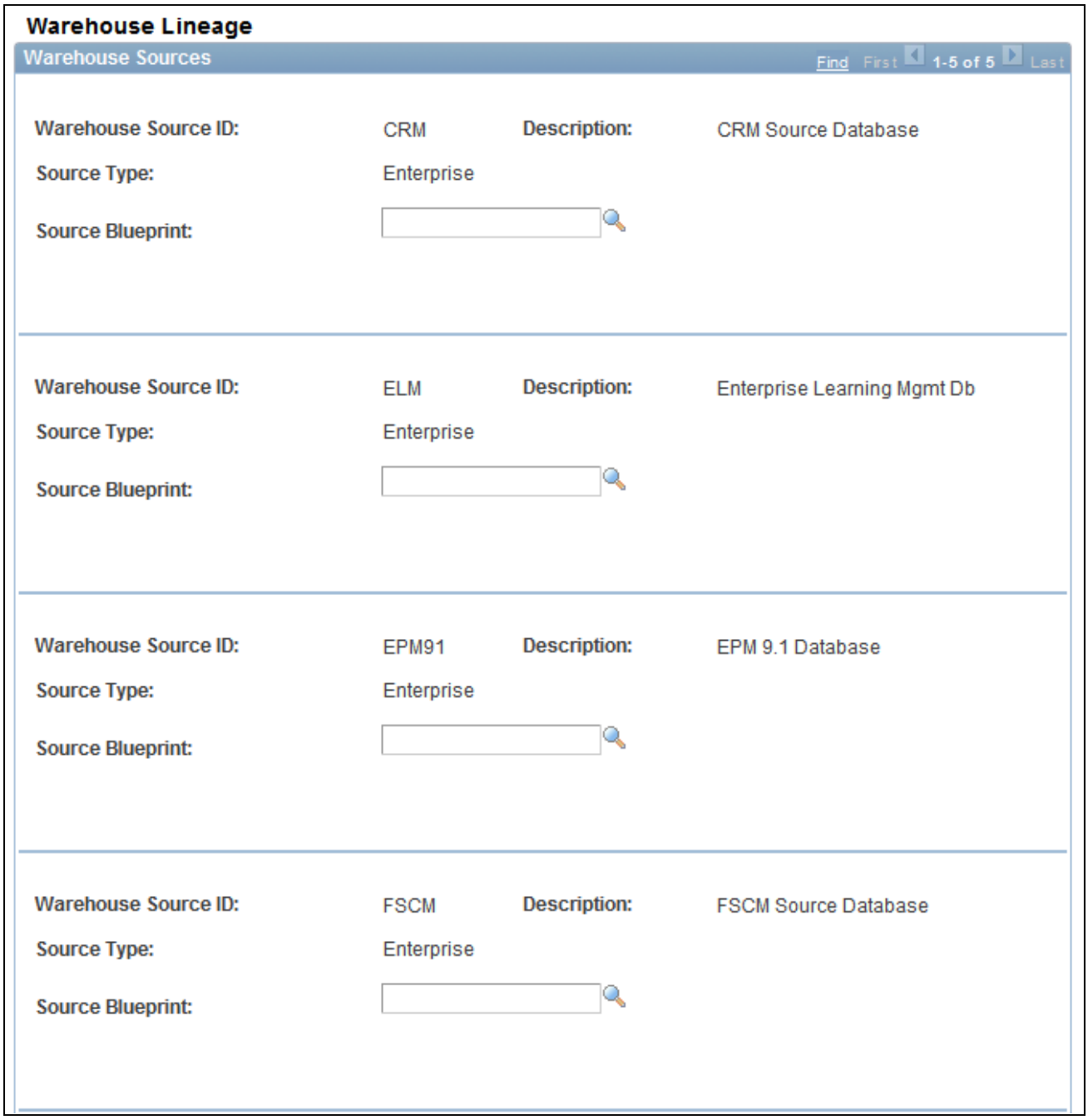
If you have customized your ETL jobs or added rows or columns to your tables, you may need to modify this field.

Warehouse Table Displays the warehouse table (OWE or MDW) associated with the preceding OWS staging table.

If you have customized your ETL jobs or added rows or columns to your tables, you may need to modify this field.

Defining Warehouse Lineage

Access the Warehouse Lineage page (EPM Foundation, EPM Setup, Warehouse Sources and Bus. Units, Warehouse Lineage).



Warehouse Lineage page

Use the Source Blueprint field to associate a blueprint with each PeopleSoft source you are using with EPM. This process define the lineage between source and warehouse tables for each PeopleSoft source.

Note. The blueprints you select for the *Source Blueprint* field are compatible with all supported source releases, regardless of the release number associated with the blueprint.

Establishing Warehouse Business Units Using the Business Unit Creation Wizard

The Business Unit Creation Wizard automates the creation of warehouse business units. Prior to running the Business Unit Creation Wizard, you must run the ETL jobs that populate the input tables used by the wizard. These jobs are collectively described as dimension mapper ETL setup jobs.

Pages Used to Run the Business Unit Creation Wizard

Page Name	*Definition Name*	*Navigation*	*Usage*
Business Unit Creation Wizard - Start	PF_BU_WIZ_START	EPM Foundation, EPM Setup, Warehouse Sources and Bus. Units, Business Unit Wizard, Business Unit Creation Wizard	Access the Business Unit Creation Wizard and start the warehouse business unit creation process.
Business Unit Creation Wizard - Review Set Controls from Source	PF_BU_WIZ_STEP_1	Click Start on the Business Unit Creation Wizard - Start page.	Review the set controls from your PeopleSoft source systems.
Business Unit Creation Wizard - Address Set Control Collisions	PF_BU_WIZ_STEP_2	Click Next on the Business Unit Creation Wizard - Review Set Controls from Source page.	Review and resolve name collisions between and among incoming and existing warehouse set controls.
Business Unit Creation Wizard - Validate Mapping	PF_BU_WIZ_STEP_3B	Click Next on the Business Unit Creation Wizard - Address Set Control Collisions page.	Review source to warehouse set control mappings and correct any set controls that collide in the warehouse.

Accessing the Business Unit Creation Wizard

Access the Business Unit Creation Wizard - Start page (EPM Foundation, EPM Setup, Warehouse Sources and Bus. Units, Business Unit Wizard, Business Unit Creation Wizard).

Use this page to access and start the Business Unit Creation Wizard.

Reviewing Incoming Set Controls, Business Unit Wizard Step 1

Access the Business Unit Creation Wizard - Review Set Controls from Sources page (Click Start on the Business Unit Creation Wizard - Start page.).

This page enables you to review the set controls from all of your sources. Set control includes both source setIDs and source business units. Review this list to make sure that you see the business units and setIDs that you expect from each source. If business units are missing, you may not have run the setup ETL jobs properly. Check the ETL error logs and the Business Unit Wizard input tables mentioned above for potential problems. Assuming that you see the business units and setIDs you expect, click Next to proceed to the Business Unit Creation Wizard - Address Set Control Collisions page.

Addressing Collisions Between Set Controls, Business Unit Wizard Step 2

Access the Business Unit Creation Wizard - Address Set Control Collisions page (Click Next on the Business Unit Creation Wizard - Review Set Controls from Source page.).

This page enables you to review name collisions between and among incoming and existing warehouse set controls. You should resolve the name conflicts by renaming the incoming set control unless you have a good reason to merge them. If you are adding a new database to an existing warehouse installation, this page gives you the opportunity to inspect name collisions between new, incoming set controls and those already in the warehouse.

You can allow colliding set controls to merge with existing set controls. However, note that an existing set control has setID assignments on warehouse record groups at the time of its creation, and these assignments may not be consistent with those of the incoming set control. This increases the chances for setID conflicts. The Business Unit Creation Wizard can fix these problems by reassigning warehouse setIDs, but if data already exists in the warehouse, then some dimensions may have to be reloaded. It is preferable to rename conflicting set controls to avoid these problems. Once you have renamed conflicting set controls, click Next to proceed to the Business Unit Creation Wizard - Validate Mapping page. It may take some time to proceed to the next page as the system processes all the inputs and validates the mapping configuration.

Validating the Mapping, Business Unit Wizard Step 3

Access the Business Unit Creation Wizard - Validate Mapping-Conflicts page (Click Next on the Business Unit Creation Wizard - Address Set Control Collisions page.).

This page displays a report based on your source to warehouse set control mappings and the analysis of setID assignments for all incoming set controls. More specifically, the report displays any instance of setID splitting (one-to-many SetID mappings) and total numbers of setIDs created and merged. If all of the set controls have been made unique as suggested, the potential for conflicts should be minimized.

If conflicts exist, you have two choices:

- Let the system correct the conflicts: Click the Accept Proposed SetIDs and Save button to automatically reassign the setIDs used by warehouse business units on various warehouse record groups in order to make the configuration valid. The system will attempt to eliminate conflicts by reducing the number of warehouse setIDs in order to eliminate one-to-many mappings.

- Modify business unit mapping or setID assignments manually: You can analyze the report to where setID splitting is occurring, and remove the conflicts by either remapping business units or choosing different SetID assignments for warehouse business units.

Regardless of how you fix the problem, if conflicts are detected, you must return to the business unit mapping page and validate the configuration again, to make sure that all conflicts have been removed. Continue to iterate the validation process until you see no more conflicts.

See Chapter 14, "Importing Source Business Units into EPM to Create Warehouse Business Units," Defining Collision Mappings (Optional), page 315.

Once any conflicts are resolved, the process of creating warehouse business units and assigning setIDs is complete. You only need to inspect the properties of the new warehouse and general ledger business units.

See Chapter 14, "Importing Source Business Units into EPM to Create Warehouse Business Units," Reviewing Warehouse and General Ledger Business Unit Creation, page 314.

Establishing Warehouse Business Units Manually

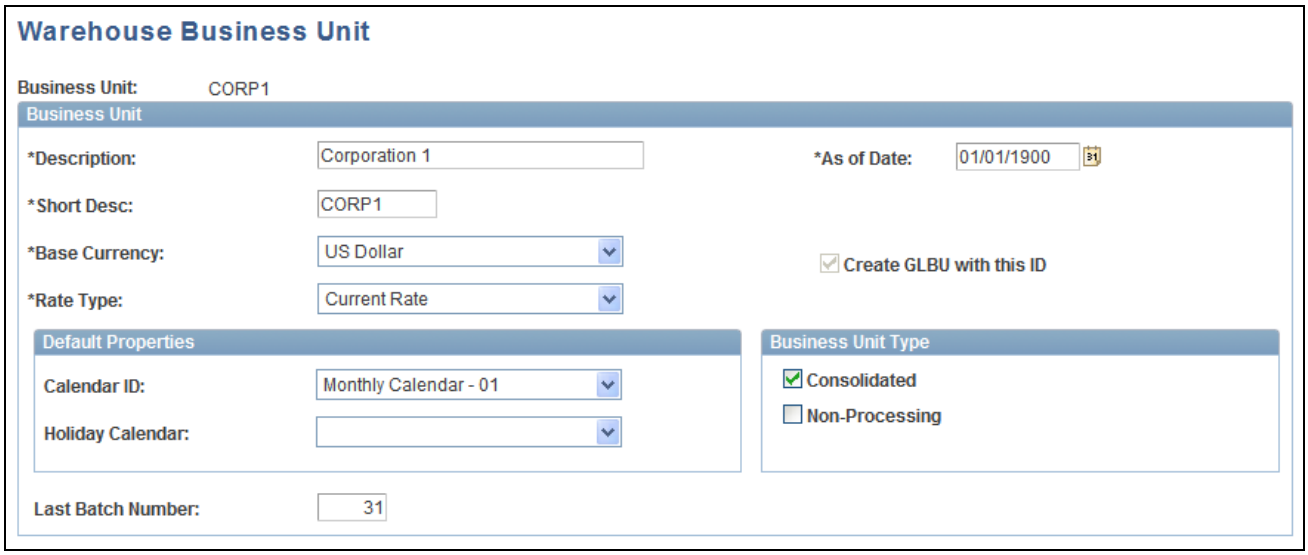
You can manually create warehouse business units by defining warehouse business units, assigning warehouse setIDs to warehouse business units, mapping source business units to warehouse business units, and then validating those mappings.

Pages Used to Create Warehouse Business Units Manually

Page Name	*Definition Name*	*Navigation*	*Usage*
Warehouse Business Unit	BUS_UNIT_TBL_PF1	EPM Foundation, Business Metadata, Business Framework, Warehouse Business Units, Warehouse Business Unit	Define or modify a warehouse business unit and its default properties.
Business Unit Mapping	PF_BU_MAPPER	EPM Foundation, EPM Setup, Warehouse Sources and Bus. Units, Business Unit Mapping	Map source set controls to warehouse set controls.
Validate Mapping	PF_BU_MAP_REPORT	Click Validate on the Business Unit Mapping page.	Validate your source to warehouse set control mappings and correct potential set control collisions.

Defining Warehouse Business Units

Access the Warehouse Business Unit page (EPM Foundation, Business Metadata, Business Framework, Warehouse Business Units, Warehouse Business Unit).



Warehouse Business Unit page

Base Currency Select the base, or primary, currency for the selected warehouse business unit.

A warehouse business unit can have *only one* base currency. This is usually the local currency for the organization, but accounting rules or other circumstances might require a different base currency.

Rate Type Select a rate type for the selected warehouse business unit.

Create GLBU with this ID Select this check box to create a general ledger business unit with the same ID as the selected warehouse business unit.

Last Batch Number enter the last batch number that was assigned.

This number automatically increments as you run batches of transactions and you should not normally need to edit it. For example, you might want to enter a batch number when you install the system for the first time; however, you only need to reset it to reuse or skip batch numbers.

Default Properties

Calendar ID Specify the default calendar type for the selected warehouse business unit.

You can choose *12*, *2*, *DR*, or *Monthly*.

The calendar ID you select appears as the default for the business unit on subsequent pages.

Holiday Calendar Specify the default holiday calendar type for the selected warehouse business unit.

You specify a holiday calendar type only if you use one of the applications for the financial services industry (PeopleSoft Risk-Weighted Capital, Funds Transfer Pricing, or Asset Liability Management).

Business Unit Type

Consolidated Select this check box to indicate that data for this warehouse business unit should be rolled up to higher level units in a business unit tree.

Non-Processing Select this check box to create a warehouse business unit without stored set control values.

Mapping Source to Warehouse Set Controls

Access the Business Unit Mapping page (EPM Foundation, EPM Setup, Warehouse Sources and Bus. Units, Business Unit Mapping).

Source ID Displays the source from which the source set control originates.

Source Set Control Displays the source set control.

Map To Indicates the direction of the set control mapping (source to warehouse).

Warehouse Set Control Select a warehouse set control that you want to map to your source set control.

Validate Click to validate your source to warehouse set control mappings and access the Validate Mapping page.

Validating Your Business Unit Mappings

Access the Validate Mapping - Conflicts page (Click Validate on the Business Unit Mapping page).

This page displays a report based on your source to warehouse set control mappings and the analysis of setID assignments for all incoming set controls. More specifically, the report displays any instance of setID splitting (one-to-many SetID mappings) and total numbers of setIDs created and merged. If all of the set controls have been made unique as suggested, the potential for conflicts should be minimized.

If conflicts exist, you have two choices:

- Let the system correct the conflicts: Click the Accept Proposed SetIDs button to automatically reassign the setIDs used by warehouse business units on various warehouse record groups in order to make the configuration valid. The system will attempt to eliminate conflicts by reducing the number of warehouse setIDs in order to eliminate one-to-many mappings.
- Modify business unit mapping or setID assignments manually: You can analyze the report to where setID splitting is occurring, and remove the conflicts by either remapping business units or choosing different SetID assignments for warehouse business units. Click the Return to Business Unit Mapper button to return to the Business Unit Mapping page.

Regardless of how you fix the problem, if conflicts are detected, you must return to the Business Unit Mapping page and validate the configuration again, to make sure that all conflicts have been removed. Continue to iterate the validation process until you see no more conflicts.

See Chapter 14, "Importing Source Business Units into EPM to Create Warehouse Business Units," Defining Collision Mappings (Optional), page 315.

Once any conflicts are resolved, the process of creating warehouse business units and assigning setIDs is complete. You only need to inspect the properties of the new warehouse and general ledger business units.

See Chapter 14, "Importing Source Business Units into EPM to Create Warehouse Business Units," Reviewing Warehouse and General Ledger Business Unit Creation, page 314.

Reviewing Warehouse and General Ledger Business Unit Creation

After you complete your warehouse business unit creation, you should review your warehouse business units and general ledger business units to ensure certain properties (such as base currency) meet your requirements.

If you created your warehouse business units automatically using the Business Unit Creation Wizard, some of your warehouse business units may have been created with a base currency or rate type that differs from the defaults that you defined for your source. If this is the case, you need to change these settings for the appropriate business units. In addition, the wizard does not associate calendars to business units. You must do this manually for all your warehouse business units.

Regardless of whether you used the Business Unit Creation Wizard to create your warehouse business units or did so manually, you should verify the properties of your general ledger business units (GLBU). PeopleSoft general ledger business units are extracted from your source system and populated in the EPM database using ETL jobs.

Pages Used to Review Your Warehouse and General Ledger Business Units

Page Name	*Definition Name*	*Navigation*	*Usage*
Warehouse Business Unit	BUS_UNIT_TBL_PF1	EPM Foundation, Business Metadata, Business Framework, Warehouse Business Units, Warehouse Business Unit	Review and modify warehouse business units and the default properties.
General Ledger	BUS_UNIT_TBL_GL1	EPM Foundation, Business Metadata, Business Framework, General Ledger Business Units, General Ledger	Review and modify general ledger business units and the default properties.

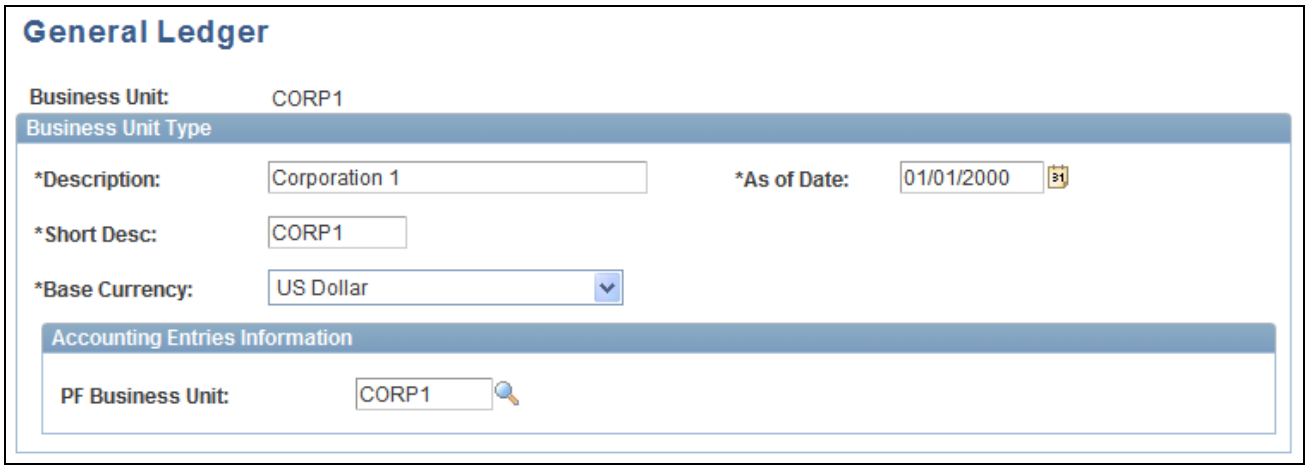
Reviewing Your Warehouse Business Units (Business Unit Creation Wizard Only)

Access the Warehouse Business Unit page (EPM Foundation, Business Metadata, Business Framework, Warehouse Business Units, Warehouse Business Unit).

If you created your warehouse business units automatically using the Business Unit Creation Wizard, review the Base Currency and Rate Type properties to ensure that they match those defined in your source system. Also, define the calendar properties for the warehouse business unit.

Reviewing Your General Ledger Business Units

Access the General Ledger page (EPM Foundation, Business Metadata, Business Framework, General Ledger Business Units, General Ledger).



General Ledger page

Review the Base Currency and PF Business Unit properties to ensure that they match the properties defined in your source system.

Defining Collision Mappings (Optional)

When you validate your source to warehouse set control mappings, you can choose to allow two tablesets merge in the warehouse. However, if two tablesets merge it is possible for collisions to occur between business keys (such as supplier ID). If such collisions do occur, you must create a collision map to resolve the collision.

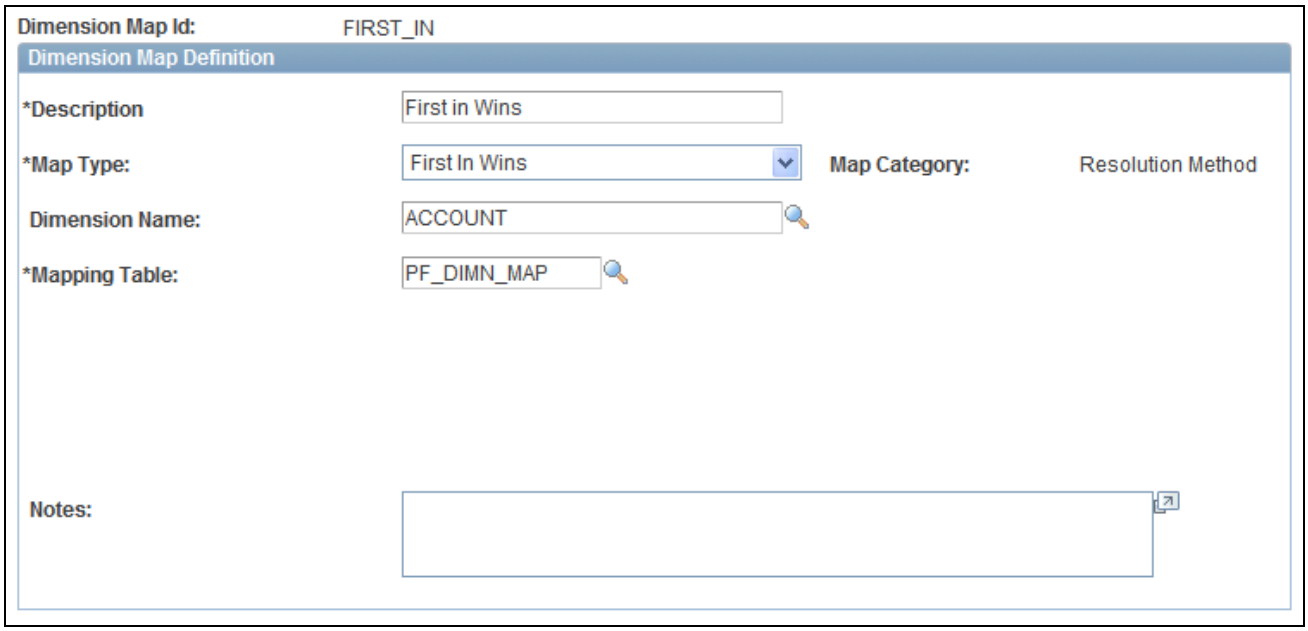
Pages Used to Define Collision Maps

Page Name	*Definition Name*	*Navigation*	*Usage*
Collision Map Definition	PF_COLLISION_MAP	EPM Foundation, EPM Setup, ETL Map Parameters, Collision Map Definition	Define collision maps to resolve business ID collisions.

Page Name	Definition Name	Navigation	Usage
Collision Map Error Report	PF_COLLISION_RPT	EPM Foundation, EPM Setup, ETL Map Parameters, Collision Map Error Reports, Collision Map Error Report	Displays the results of the error out duplicates collision map.

Defining Collision Maps

Access the Collision Map Definition page (EPM Foundation, EPM Setup, ETL Map Parameters, Collision Map Definition).



Collision Map Definition page

Map Type Select a collision map type to resolve any business ID conflict resulting from merging tablesets.

You can select:

First In Wins: The first instance of a business ID is loaded into a tableset and subsequent instances of the same ID from different tablesets are ignored.

Error Out Duplicates: The first instance of a business ID is loaded into a tableset, but subsequent instances of the same ID from different tablesets are sent to an error table where they can be inspected and remapped. You can examine these errors by accessing the Collision Map Error Reports page.

If you select this option, the Warehouse Record and Dimension ID fields appear for editing.

Use Mapping Table: Each incoming business ID is checked against a mapping table. If an entry does not exist in the mapping table, the row errors out.

Dimension Name	Select the target dimension that contains the merging tablesets. The list is drawn from the set of dimensions defined in the Define Dimension page.
Mapping Table	Select a mapping table that stores resolved conflicts to verify future incoming business IDs. You can use the default mapping table (PF_DIMN_MAP) or create your own mapping table in Application Designer.
Warehouse Record	Select the OWE or MDW table that associates with the dimension selected for the Dimension Name field. This table is used for prompting in the Error-Out Duplicates map type.
Dimension ID Field	Select the ID column for the table you specified in the Warehouse Record field.
Description Fieldname	Select the column that stores the description for the table you specified in the Warehouse Record field.

Reviewing the Error Out Duplicates Collision Map Error Report

Access the Collision Map Error Reports page (EPM Foundation, EPM Setup, ETL Map Parameters, Collision Map Error Reports, Collision Map Error Report).

This page displays the results of the error out duplicates collision map, where the first instance of a business ID is loaded into the relevant tableset and the subsequent instances of the same ID from different tablesets are sent to this page for review. For each row of data you can chose to:

- *Accept:* The mapping between the colliding business IDs is preserved as displayed.
- *Create:* You can create a new business ID that can be loaded into the applicable tableset.
- *Defer:* You can defer action to a later time.
- *Select:* You can select an existing member of the tableset as a mapping for this incoming business ID.

If you select *Create* or *Select*, the Target Business ID field becomes available for editing and you can create or select another business ID to load into the applicable tableset.

Chapter 15

ETL Configurations

This chapter discusses how to perform the following *optional* ETL configurations:

- Create Master Sequencer jobs
- Use the Master Run Utility to automatically run ETL jobs
- Support UniCode data
- Convert incremental load jobs to destructive load jobs
- Add new Environmental Variable

Using the Master Sequencer Utility to Create Master Sequencer Jobs

The master sequencer utility (Create_MasterSequence) enables you to create master sequencer jobs using delivered sequencer jobs as input. You will recall that sequencer jobs invoke and run other jobs. You can create a master sequencer job to invoke and run all the required sequencers that populate a specific fact table , or populate all fact tables for a given data mart, or all fact tables for an entire EPM warehouse.

For example, you can create a master sequencer job to automatically invoke and run 20 sequencer jobs from the Receivables mart. To do so, simply use the master sequencer utility to specify the 20 jobs you want to invoke, enter the applicable job parameters, and import the generated *.dsx file into your project.

The master sequencer utility is available in the following DataStage folder: *Utilities\Job_Utils\MasterSequence\Create_MasterSequence.*

Creating a Master Sequencer Job

To create a master sequencer job using the master sequencer utility:

1. Identify the individual sequencer jobs you want to include in the master sequencer job.

   The ETL Lineage appendix can help you determine which ETL sequencer jobs are required based on your needs.

   See Appendix D, "Using the PeopleSoft EPM Lineage Spreadsheets," page 673.

   Note. The master sequencer utility uses only sequencer jobs as input; please do not use server jobs as input. Also, the sequencer jobs you select should be compiled and ready for use.

2. Copy the list of sequencer jobs and paste the names into a text file.

   Ensure that each job is entered on a separate line. For example,

   ```
   SEQ_J_Dim_PS_D_DT_PATTERN_OWE
   SEQ_J_Dim_PS_D_BUS_UNIT_PF_OWE
   SEQ_J_Dim_PS_D_BUSINESS_UNIT_EX
   ```

   Note. The master sequencer job uses the text file as an input and reads the list of sequencer jobs from the file.

3. Save the text file and specify the input file path (SRC_JOBPATH).

   The file path you specify should be local to the server; do not specify a path to a client machine.

   Note the file path of the text file for later use.

4. In DataStage Director, navigate to the Master Sequencer Utility using the following navigation: *Utilities\Job_Utils\MasterSequence\Create_MasterSequence.*

5. Select *Create_MasterSequence* and run.

   The *Create _MasterSequence - Job Run Options* window appears where the job input parameters are displayed.

6. In the Parameters tab, use the *Enter Source text file Name* field to enter the file path for the text file you created in step three.

7. Use the *Enter Target File Path* field to enter a location where you want the new master sequencer job (the output *.dsx file) stored.

   The file path you specify should be local to the server; do not specify a path to a client machine.

   The file path must have the **.dsx* extension appended to it, for example, *C:\MSEQ\Mseq01.dsx.*

8. Use the *Enter the Jobname* field to enter the name of the new master sequencer job.

9. Use the *Choose the Master Sequence Type* field to specify the mode in which you want the sequencer jobs to be run.

   Valid values are *Parallel* or *Sequential.*

   Note. Ensure you select the proper mode for the sequential jobs included in the master sequencer. Only staging sequencer jobs can be run in parallel. Other dimension and fact jobs may have dependencies and, unless those dependencies are identified and managed accordingly, we advise that you do not run them in parallel.

10. Click the Run button to generate the master sequencer job.

    This process generates the master sequencer job and stores it in the target file path you specified in step seven.

11. Check the log file of the *Create_MasterSequence* job for warning messages

    Hard-coded values in the sequencer job parameters are listed as warning messages in the log file. If you encounter a warning message, correct the master sequencer job parameters accordingly, and re-run the *Create_MasterSequence* job.

12. Import the generated master sequencer job into your DataStage project.

    If a master sequencer job with the same name already exists in the project, it is overwritten and placed in the same category.

    If the master sequencer job does not exist in the project, it is placed in the following default location: *Utilities\Job_Utils\MasterSequence.*

    Once the master sequencer job is imported into your project, you can move it to another category.

    The utility does not provide annotations for the master sequencer job, but you can provide annotations if you wish.

Note. You must compile the master sequencer job before you can use it.

See Chapter 13, "Running Initial Setup Jobs," Compiling ETL Jobs, page 276.

Using the Master Run Utility to Automatically Run Your ETL Jobs

This section provides an overview of the Master Run Utility and discusses how to run the utility.

Understanding the Master Run Utility

PeopleSoft provides the Master Run Utility, which enables you to automatically run a set of jobs located in a flat file on the DataStage Server. When you use the Master Run Utility, it reads a list of jobs that are present in a specified flat file and triggers the jobs to run in serial mode, using the dependency logic specified in the Input flat file.

The following features are incorporated into the Master Run Utility:

- Run jobs from a specified flat file.
- Reset jobs and rerun when they are in an aborted stage.
- Active restart recovery.
- Run jobs in dependent or independent modes.
- Generate a job status report in the log file at the end of a run.

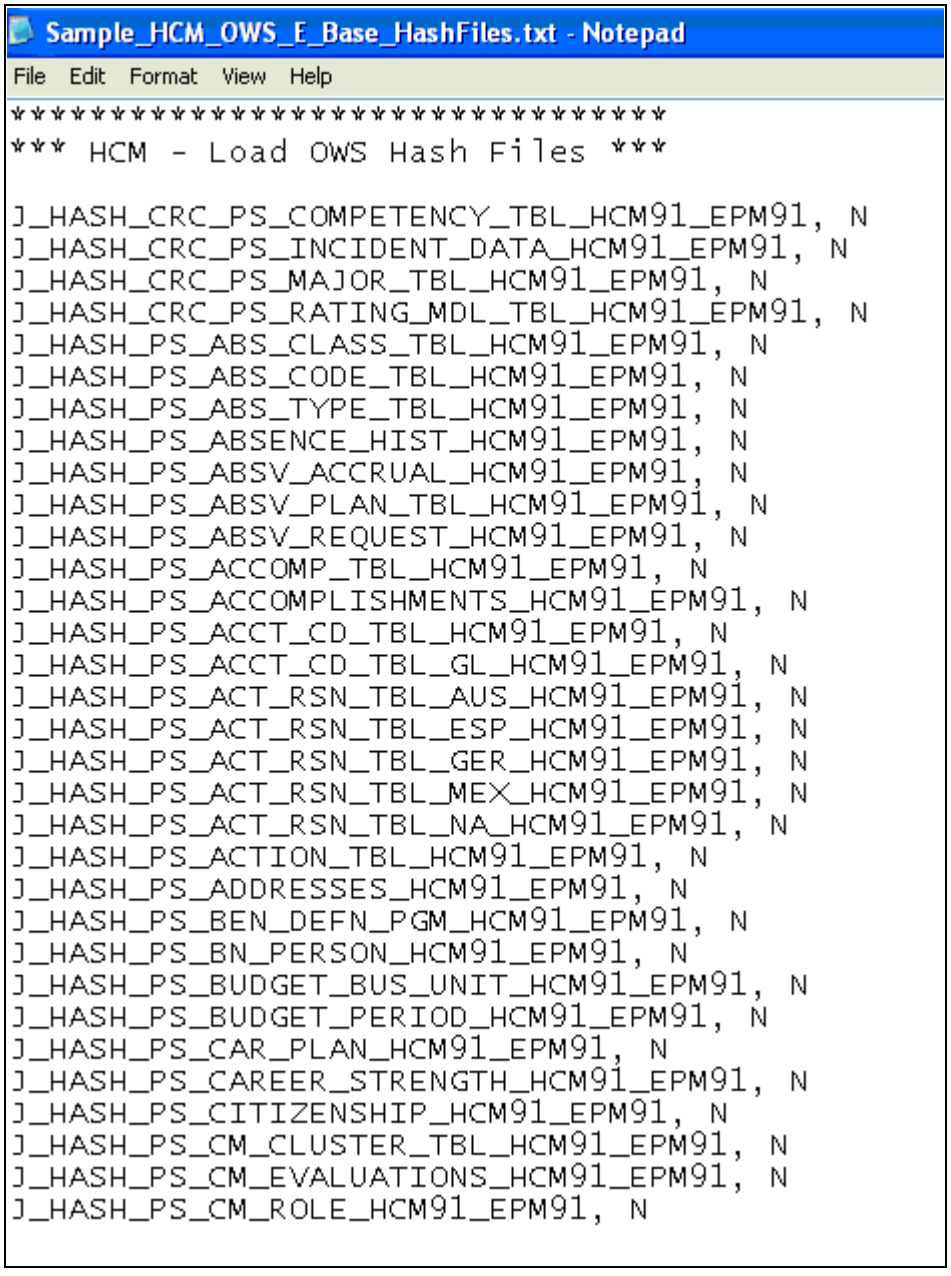
Location of the Master Run Utility

The Master Run Utility is available in the *Common_Utilities.dsx* file. After you import this .dsx file, you can locate the utility using the following navigation: *Jobs, Utilities, Job_Utils, Master_Run_Utility.*

Input File for Master Run Utility

The Input flat file contains the list of ETL jobs that are available in the DataStage Server and indicates whether a job is *independent* or *dependent.* Dependency information is used to determine job execution in the Master Run Utility. If a job labeled as independent fails to run properly, the Master Run Utility logs the information and proceeds to the next job. However, if a job labeled as dependent fails to run properly, the utility itself aborts. All jobs listed in the flat file contain the suffix *N* (Independent job) or *D* (Dependent Job) to indicate their dependency.

The Input flat file also contains comments that describe each set of ETL jobs. Comments are prefixed with an asterisk (*) to clearly indicate they are comments. The following screen shot provides an example flat file:



Sample Flat File (HCM_OWS_E_Base_HashFiles.txt)

The sample file above contains the list of HCM jobs used to load data into OWS target tables.

Restart Recovery Feature

When a dependent job fails to run properly, the Master Run Utility automatically aborts the job and all subsequent jobs. Assuming you fix the problem that caused the job to fail, you must rerun the Master Run Utility to complete the process of running those jobs that were aborted. Instead of running all the jobs in that particular folder, the restart recovery feature enables you to rerun only those jobs that were aborted.

For example, assume the file SAMPLE_HCM_E_GLOBAL_DIMENSIONS_Base_Tables.txt contains Global Dimension jobs that are all dependent to each other. If the SEQ_J_Dim_PS_D_POS job is aborted, the entire utility aborts. Later you fix the issue in the SEQ_J_Dim_PS_D_POS job. This time, you can run the Master_Run_Utility with the Restart Recovery option set to *Yes* so that it runs the jobs from SEQ_J_Dim_PS_D_POS only and ignores the jobs that have previously completed successfully.

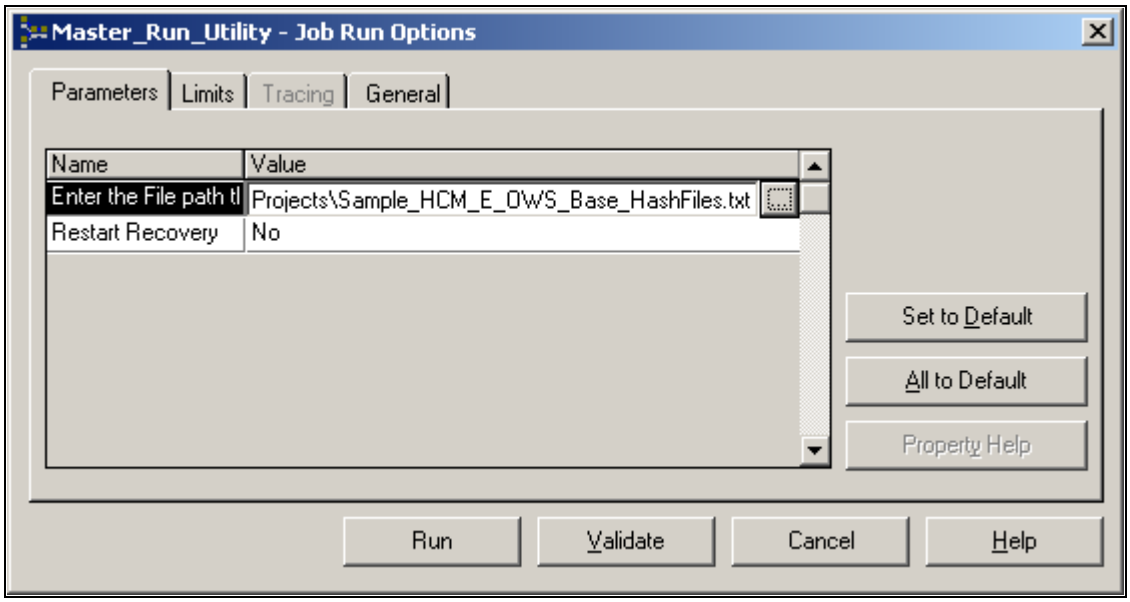
Running the Master Run Utility

Perform the following steps to run the Master Run Utility:

1. In DataStage Director, navigate to the Master Run Utility using the following navigation: *Jobs*, *Utilities*, *Job_Utils.*

2. Select *Master_Run_Utility* and click Run.

   The *Master_Run_Utility - Job Run Options* window appears.



   Master_Run_Utility - Job Run Options Window

3. Enter the path to the location to the flat file and specify whether you want to use the restart recovery feature.

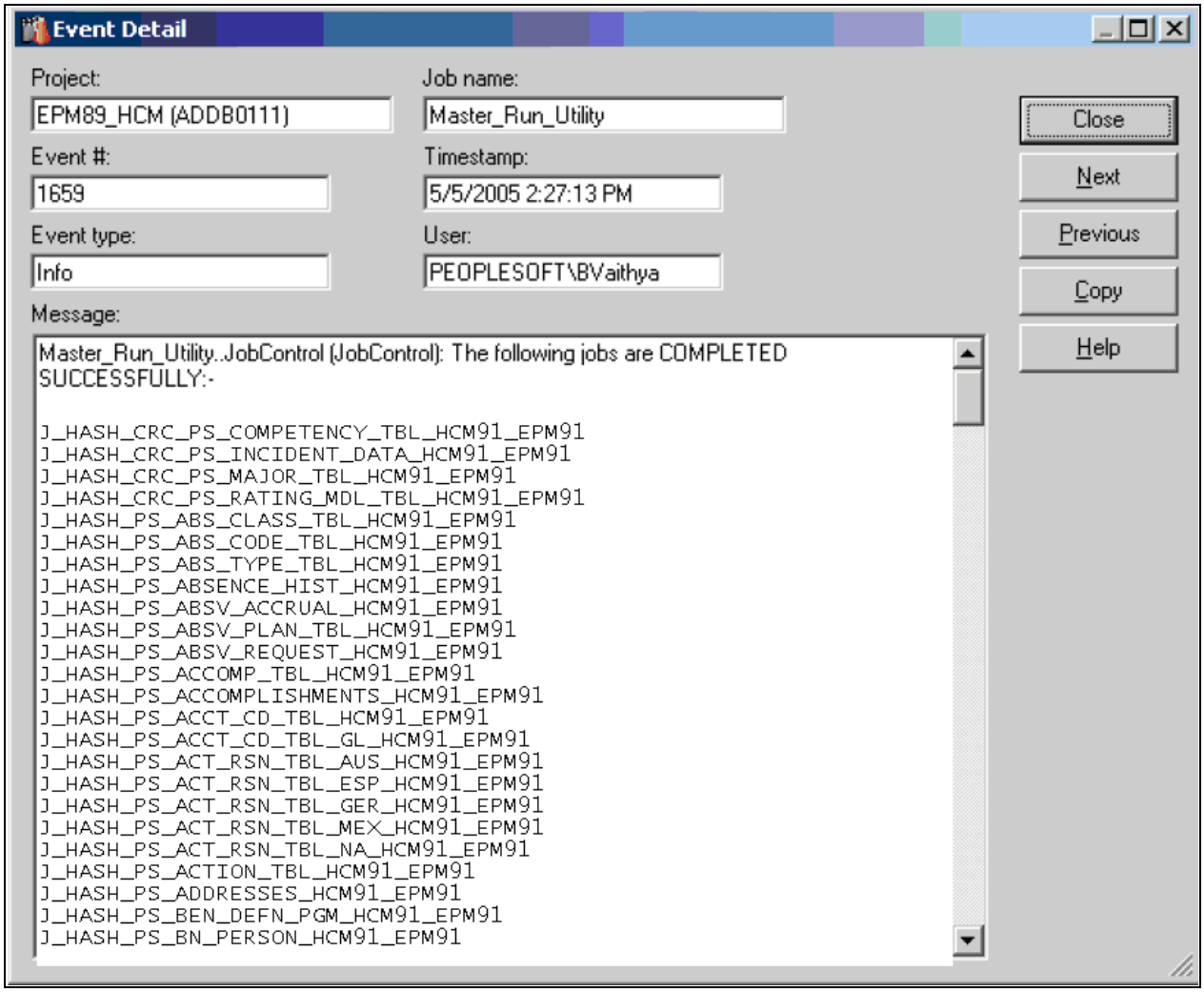
   Click Run.

4. From the DataStage Director menu, select *View, Log.*

   The Master Run Utility generates a report with the following information:

   - Jobs that have completed successfully.
   - Jobs that have completed with warnings.
   - Jobs that have aborted
   - Jobs that have not compiled
   - Jobs that have incorrect names.

5. If you double-click log entries that contain the phrase *COMPLETED SUCCESSFULLY*, the Event Details window appears and displays all the jobs that have successfully completed.



Event Details Window

Likewise, if you double-click log entries that contain the key word *ABORTED*, the Event Details window appears and displays all the jobs that have aborted.

Supporting UniCode Data

To support Unicode data, you must have first installed DataStage Server with the *Install NLS for DataStage Server* check box selected. You must then configure NLS options in DataStage Administrator.

To configure NLS options:

1. In DataStage Administrator, select the Projects tab and then the project you want to configure NLS options.

2. Select the NLS button to select NLS options.

   These options are available only if DataStage Server was installed with the NLS option.

3. In the Server Maps tab, select the correct NLS map for the project default.

   This value is used in all the jobs to map Unicode data.

4. If you do not see the correct NLS map in the list, you can click the Install>> button to view all available maps and load the particular map you require.

5. Select the NLS tab to ensure that your selected map displays as the Project Default for individual jobs in the DRS stage.

   Note. This value can be overwritten at the job level by changing this value in the DRS stage.

6. Click OK to return to the Projects tab.

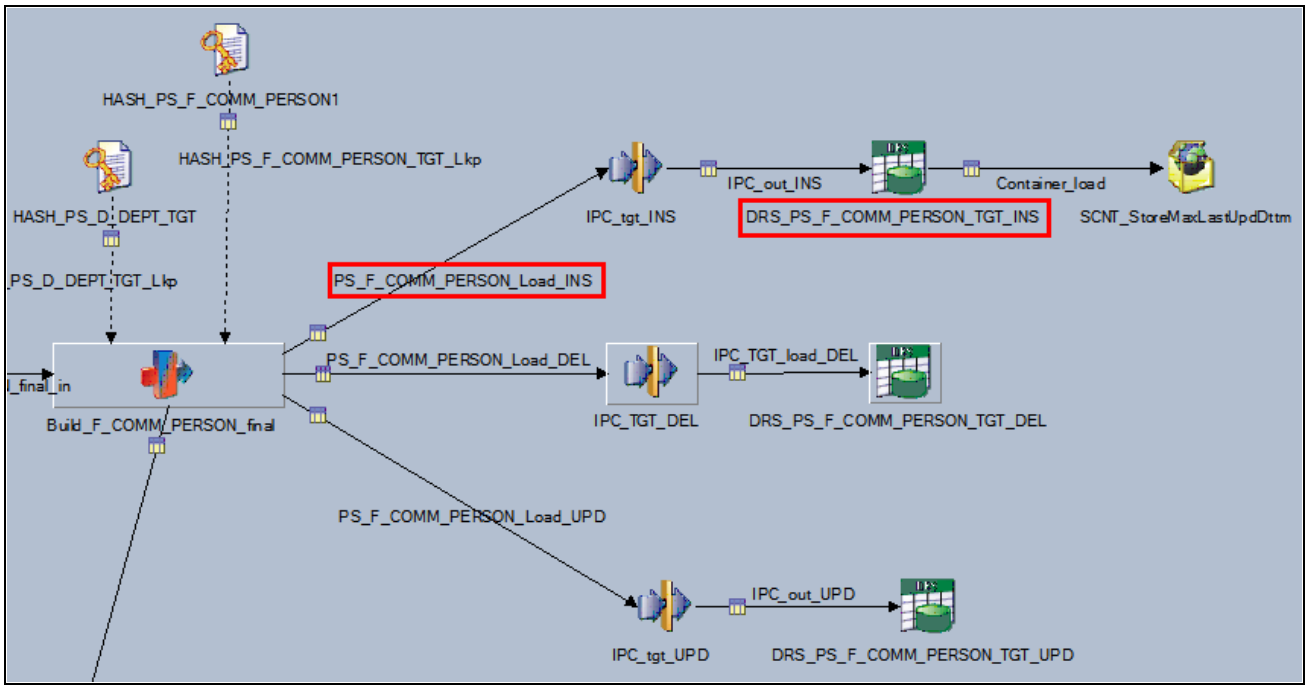
Converting Incremental Load Jobs to Destructive Load Jobs

As part of the ETL configuration process, you can convert incremental load jobs to destructive load jobs. However, because server jobs that use CRC logic would require modification to at least 80% of the design, it is better not to alter the existing jobs and create a new destructive load job from scratch.

The section below discusses how to convert incremental load jobs that use the DateTime stamp.

Converting Jobs that Use the DateTime Stamp

The changes required to convert an incremental load job (that uses the DateTime stamp) to a destructive load job can be demonstrated using the J_Fact_PS_F_COMM_PERSON job as an example.

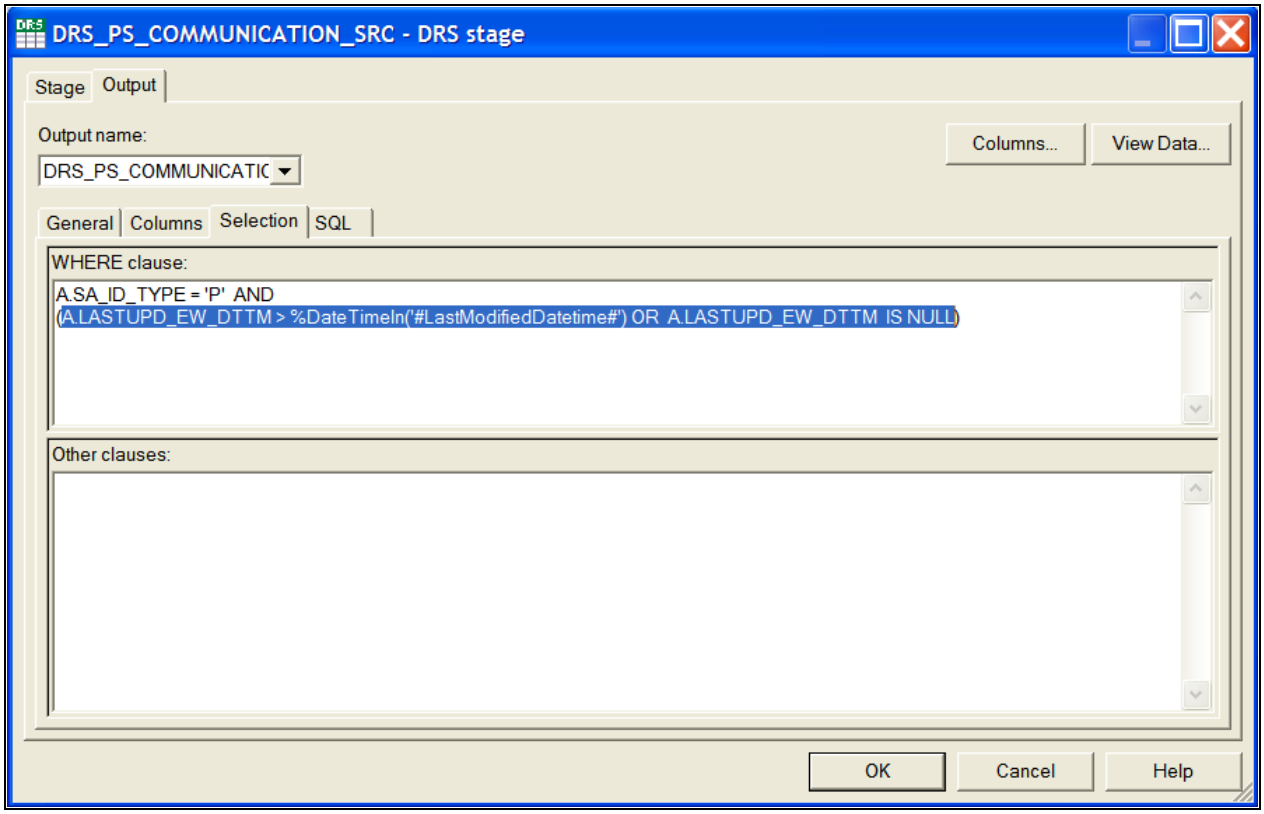


Example of the incremental load in the J_Fact_PS_F_COMM_PERSON job

To convert an incremental load job (that uses the DateTime stamp) to a destructive load job:

1. In DataStage Designer, open the server job you want to convert.
2. Open the source DRS stage and select the Output tab.

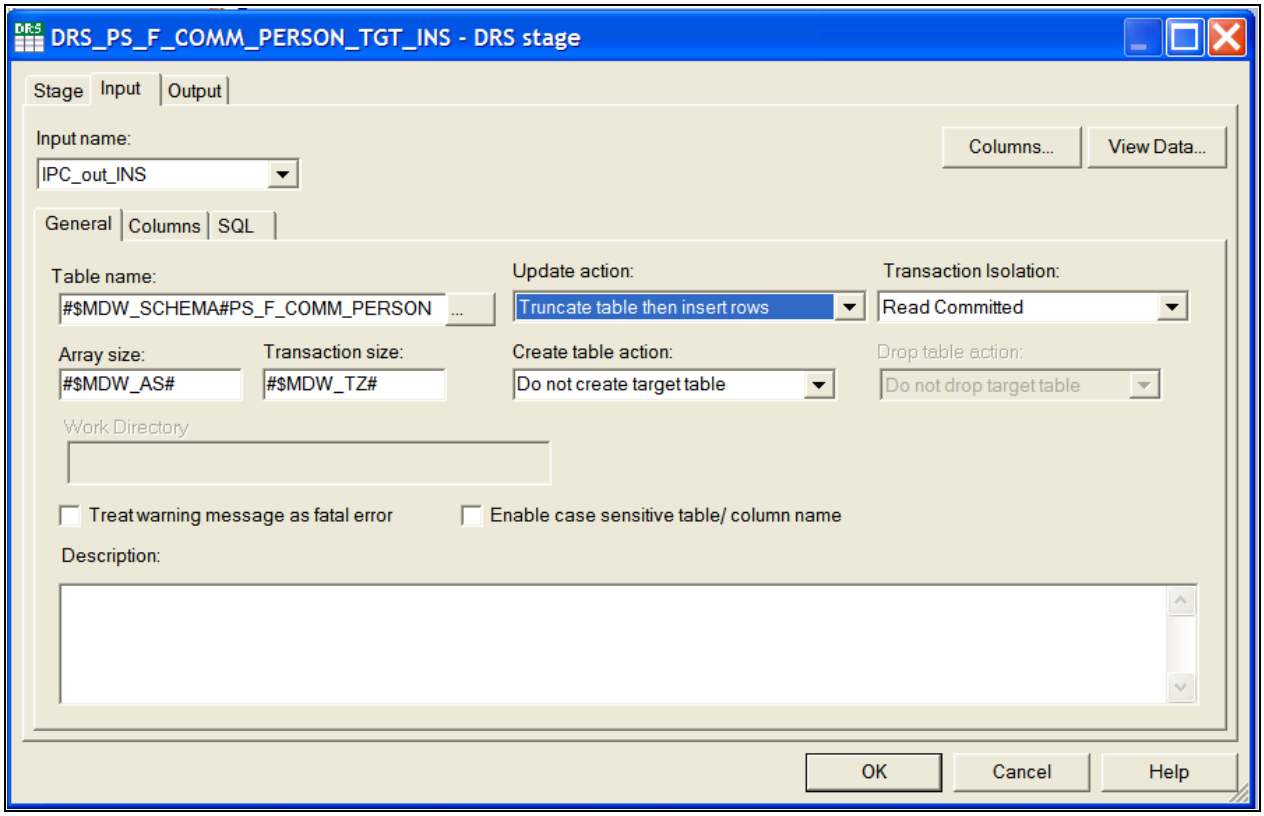
3. In the Selection sub-tab, locate the WHERE clause and delete the last update date time portion (highlighted below).



Deleting the WHERE clause

4. Click OK to save your changes.
5. Open the insert (*_INS) target DRS stage and select the Input tab.

6. In the General sub-tab, select *Truncate table then insert rows* for the Update Action field.

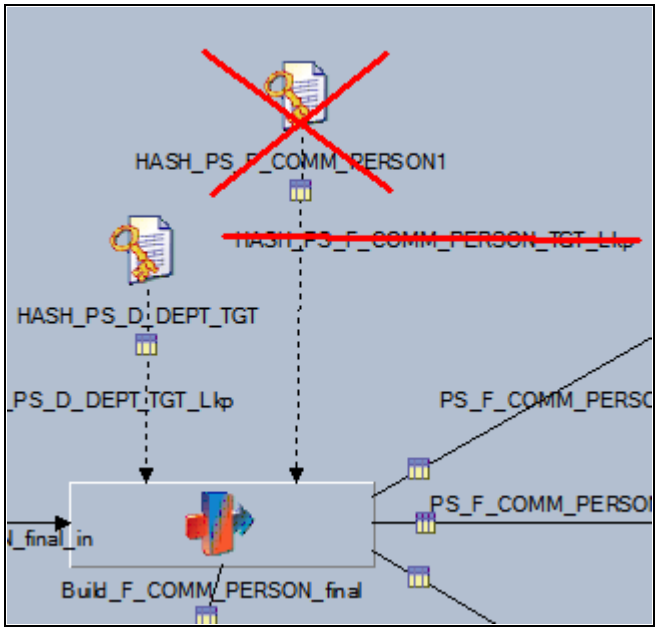


Update Action field

7. Click OK to save your changes.
8. Delete the StoreMaxLastUpdDttm container and link.
9. Delete the delete (*_DEL) target DRS stage and link.
10. Delete the update (*_UPD) target DRS stage and link.

11. Delete the hash target table lookup (the hash lookup that is performed against target table data) and link.

    Because this hash load is used to identify updated or new records and you are converting the job to destructive load, the hash load is no longer needed.



Deleting the hash target table lookup

12. Open the last transformation stage in the job (it should immediately precede the insert target DRS stage).

    New rows are identified in this stage and this is done to retain the Created_EW_DTTM of rows.

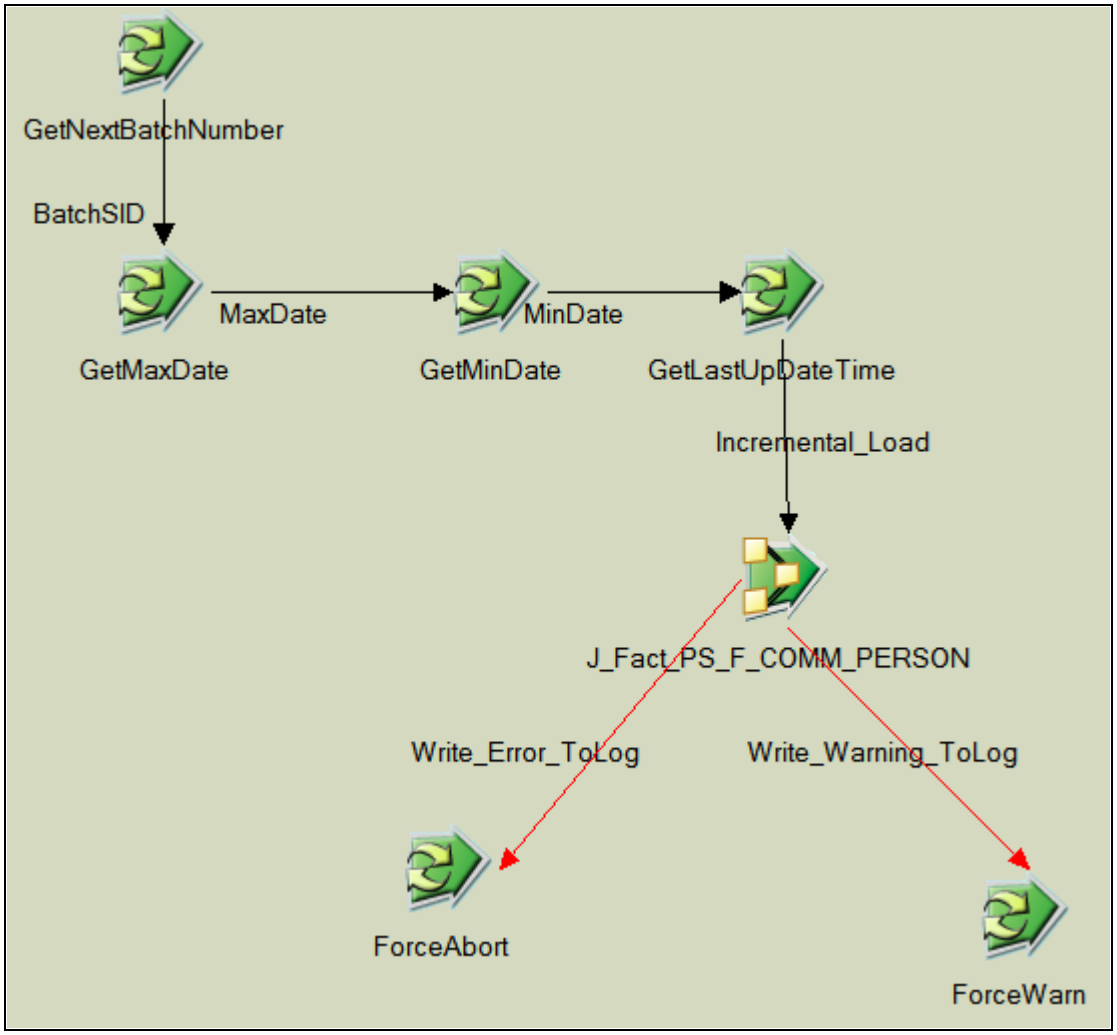
    In the example job above, the last stage is called Build_F_COMM_PERSON_final.

13. Delete the *InsertFlag* stage variable and click OK to save and exit the window.

14. Select Edit, Job Properties from the menu and change the target column value for CREATED_EW_DTTM to *DSJobStartTimestamp*, which is a DS Macro (and same as for the field LASTUPD_EW_DTTM).

15. Delete the *LastModifiedDateTime* job parameter and click OK to save and exit the window.

16. Open the corresponding sequence job that calls the server job and delete the *GetLastUpDatetime* job activity stage (which calls the routine of the same name).



Sequencer job

17. Select Edit, Job Properties from the menu and delete the *LastUpdDateTime* job parameter if it is present.

    This parameter is not present in every job.

    (I say "IF" because it may not be present in the sequence job. It is not needed in the sequencer anyway).

18. Change the job annotations and descriptions to reflect the change.

19. Save changes and exit.

20. Save and recompile the job.

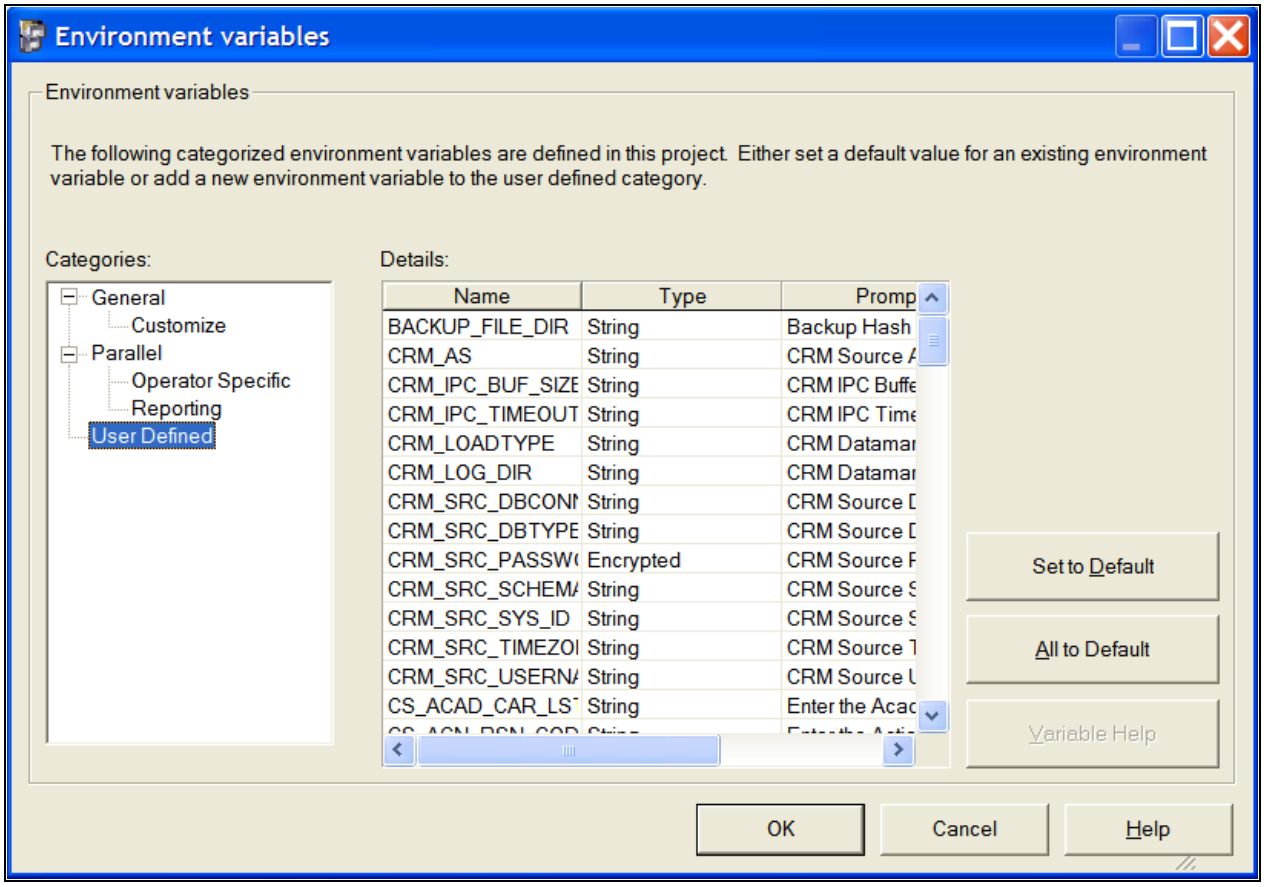
Adding New Environmental Variables

Environmental variables are project level parameters which are typically used across projects. The advantage to environmental variables is that they can be set at the project level and all associated jobs in that project will automatically use the value.

Creating a New Environmental Variable

To add a new environmental variable:

1. In DataStage Administrator, select the Projects tab.
2. Click the Properties button and select the General tab.
3. Click the Environment button and select User Defined from the Categories pane.



Defining Environmental Variables

4. Enter your new environmental variable parameters for the following fields:
   a. *Name*
   b. *Type*
   c. *Prompt*
   d. *Value*
5. Click OK to save your changes.

Adding an Environmental Variable to a Server Job

To add the new environmental variable to a server job:

1. In DataStage Designer, select Jobs from the project tree.
2. Select the job you want to add the environmental variable.
3. Select Edit, Job Properties from the menu and select the Parameters tab.
4. Click the Add Environment Variable... button.

   The Choose Environment Variable window appears.
5. Select the new environmental variable from the Choose Environment Variable window.
6. Click OK to save your changes.

   The new environmental parameter is now a part of the job parameters.

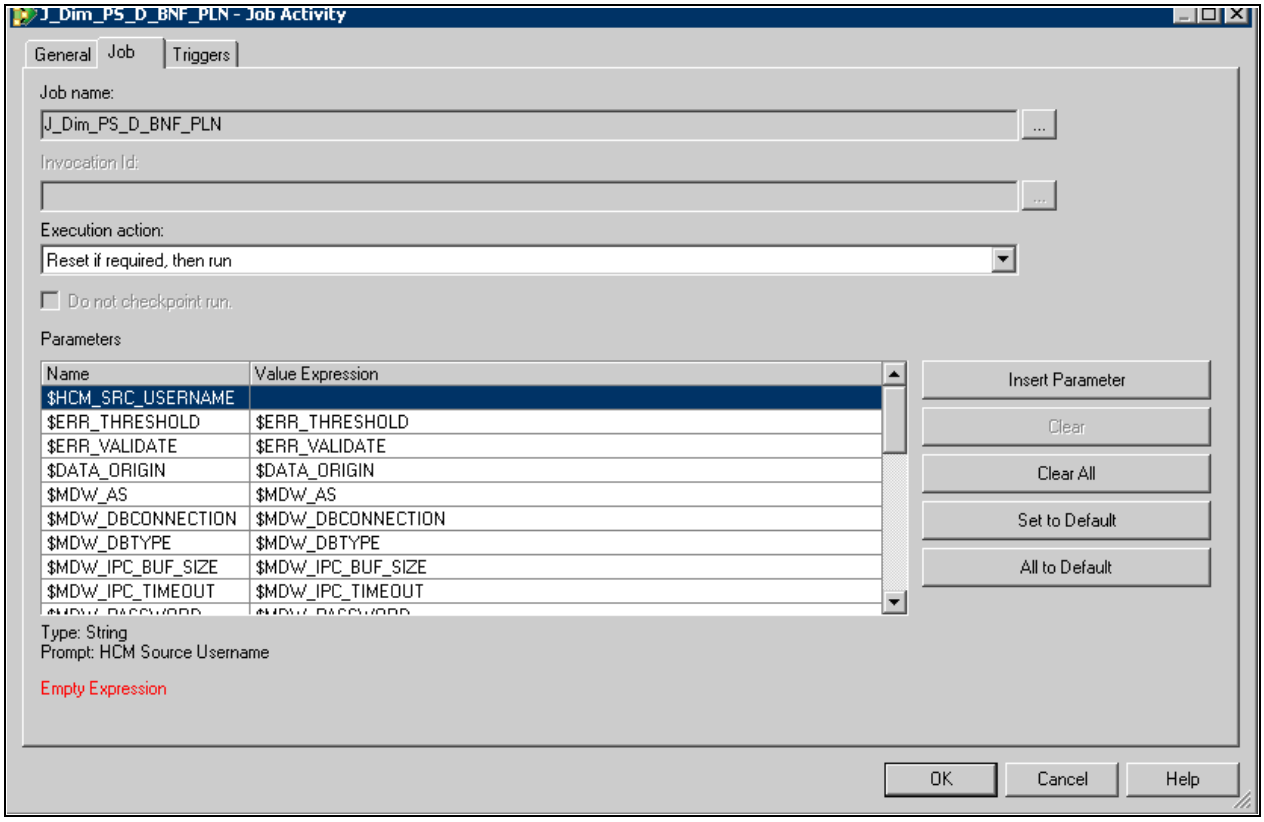
   The Default value can be changed to $PROJDEF in the job parameters to signify that the value shall be taken from the project default value unless overwritten.
7. Save and recompile the job.

Updating the Related Sequencer Job

If there is a sequencer job that calls the server job you modified, you must update the sequencer job to add the parameter value to the job activity stage where the value is passed to the called job.

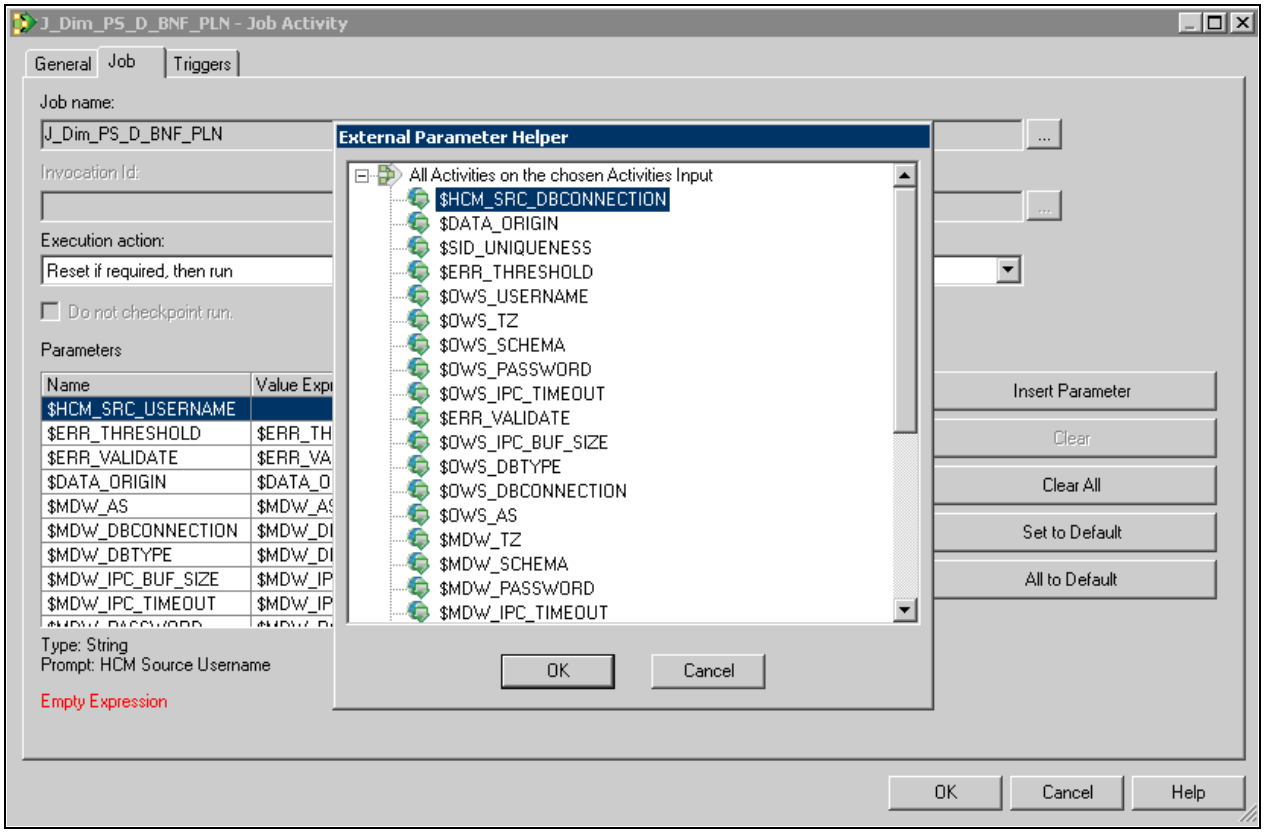
To modify the related sequencer job:

1. Select Edit, Job Properties from the menu and select the Job tab.



Job Tab of Job Properties

2. Select the parameter from the list, and click Insert Parameter.



Inserting Your New Parameter

3. Select your parameter, then click OK to save changes and exit.
4. Save and recompile the job.

Part 4

Setting Up the Operational Warehouse - Enriched for EPM Analytical Applications

Chapter 16

Setting Up and Working with Metadata for the Operational Warehouse - Enriched

This chapter provides an overview of metadata, lists common elements, and discusses how to:

- Find metadata objects.
- Apply the hidden flag to metadata objects.
- Set up record metadata.
- Set up rule metadata.
- Set up tablemaps.
- Set up datamaps.
- Set up expressions.
- Use data sets.
- Set up filters.
- Set up constraints.
- Set up metric metadata.
- Set up record summary metadata.
- Set up report metadata.
- Clone metadata.
- Use the Metadata Mover utility.
- Delete metadata.
- Create user-defined functions.

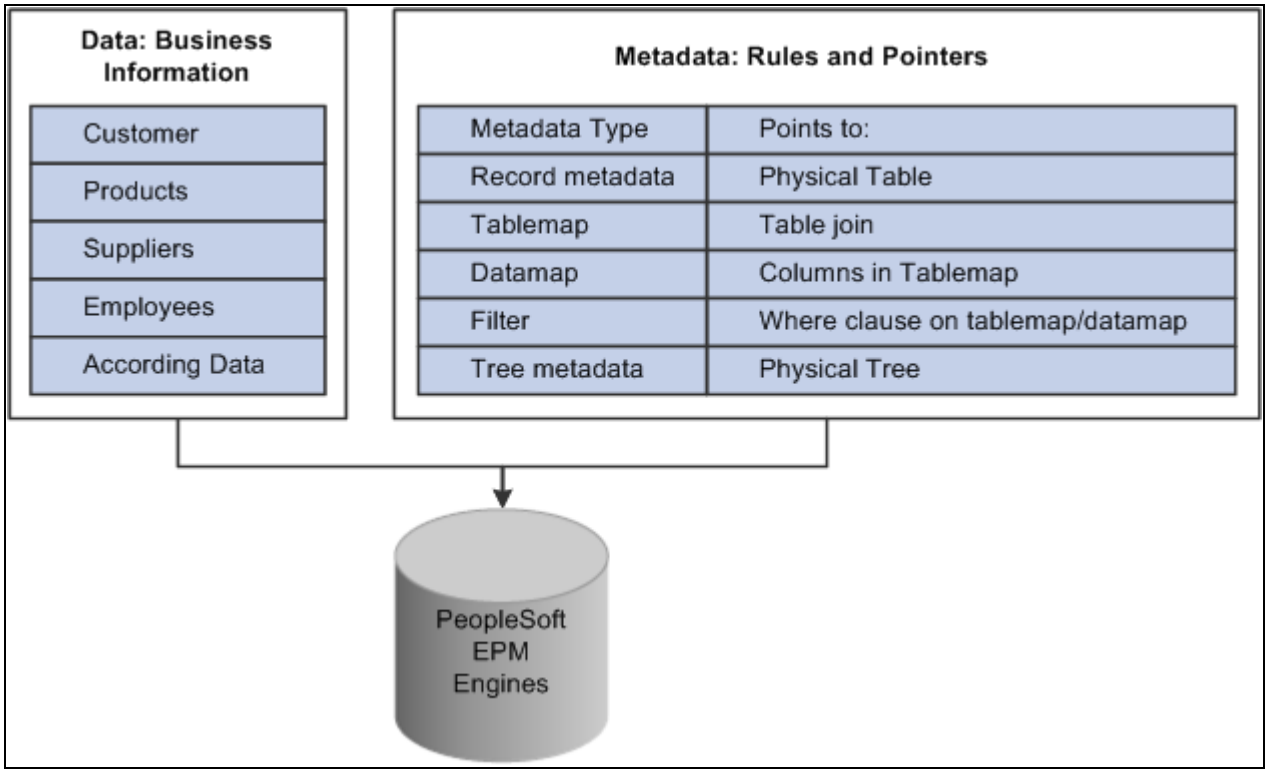
Understanding Metadata

The PeopleSoft Analytical Applications are supported by a framework of metadata. Metadata defines everything from table and data structures to rules for running processes. Metadata is central to the entire Analytical Applications product suite.

When you run PeopleSoft EPM processes, you require two types of input: your business data and the appropriate EPM metadata. Metadata provides an abstraction layer that enables technical users to establish dynamic relationships between tables, business users to easily identify the data that interests them without having to know the database structure, and administrators to manage processes.

Metadata is information that is used by processes and application engines to define rules and physical objects such as tables or trees. For instance, you may want an application engine to process certain columns from certain tables. Instead of entering a long SQL statement, you can define a datamap.

This diagram shows the relationship between data and metadata in EPM.



EPM data and metadata

This section discusses:

- Metadata terms and objects.
- SQL object ID.
- Metadata utilities.

Metadata Terms and Objects

When we discuss metadata, we discuss:

- Technical metadata.

  Technical metadata defines the physical structures of EPM and enables users to work with EPM Analytical Applications without the need to understand the underlying structures. Examples are record metadata, tablemaps, datamaps, filters, and constraints.

- Business intelligence metadata.

  Business intelligence metadata defines how information is to be used to provide end users with the information they need to work most productively. An example of business intelligence metadata is the KPI Manager.

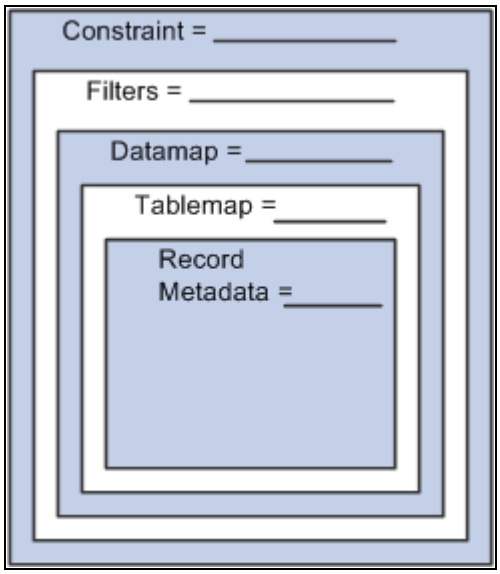
- Operational metadata.

  Operational metadata provides information about the flow of data through EPM. An example of operational metadata is extract, transform, and load (ETL) jobs.

The metadata structure that is used by the PeopleSoft EPM engines is built in several layers or levels, each dependent on the layer below it. The following table lists the metadata layers that are found in PeopleSoft EPM in order from the lowest to the highest level:

Metadata Object	***Includes***
Record metadata	Defines the lowest level of EPM metadata table objects. This is the foundation on which all other metadata is defined. Record metadata defines and identifies EPM data tables.
Tablemaps	Define the physical relationships (joins) between tables and are foundations for datamaps.
Datamaps	Built on tablemaps, datamaps are similar to table definitions in that they describe a logical view of the tables themselves. They enable you to select information from different tables that are specified in a tablemap and define it as if it were one entity or table.
Constraints	Built on datamaps and can use one or more filters to define your business processing rules.
Filters	Enable you to define what subset of data gets processed by or uses a specific business rule.

This diagram illustrates how PeopleSoft EPM metadata is nested to define a SQL statement.



EPM metadata, SQL

The resulting SQL is SELECT (Datamap) FROM (Tablemap) WHERE (Constraints/Filters).

Other types of metadata and terms are:

ERP metadata	Examples are calendars, fiscal year, accounting period, business unit, and setID. See Chapter 4, "Setting Up EPM Business Rules," page 45.
Data sets	Used as input for various engines.
Expressions	Enable you to create virtual columns that are made up of mathematical calculations based on actual fields on a table.
Tree metadata	Captures information about the trees that you have set up in PeopleSoft EPM. It is mainly used for reporting.
Rule metadata	Gathers the rules for PeopleSoft EPM engines based on the standard keys of setID, business unit, model ID, and effective date. Rule metadata is used by PeopleSoft EPM engines to recursively determine inheritance rules between models.
Balancing rules	Enable you to track before and after amounts in the system.
Job totals	Define flash totals for data to be used as input or output to the different source or target tables in the system.
PF_RECONCILIATION engine	PF_RECONCILIATION uses the job totals and balancing rules metadata that you set up to validate balances.
Engine and job metadata	Delivered metadata for running jobs and jobstreams.

Not all the metadata objects are described in this chapter. Refer to the following chapters for information about rule and engine metadata and balancing rules and job totals metadata.

See Also

Chapter 20, "Streamlining Processing with Jobstreams," Working with Engine Metadata, page 469

Chapter 20, "Streamlining Processing with Jobstreams," Setting Up Job Metadata, page 475

Chapter 21, "Setting Up and Using Profit Manager," Using Balancing and Reconciliation Features, page 528

SQL Object ID

The SQL object ID is a system-generated number that identifies a SQL object that is generated by a metadata component and is stored in the PeopleSoft SQL repository. The prefix identifies the PeopleSoft EPM product, metadata object, and sequential number for each metadata object respectively, for example PF$_MR_140. Many setup pages within PeopleSoft EPM have a SQL object ID associated with the page definition.

Various types of SQL object ID prefixes exist, depending on the metadata object that you set up:

SQL Object ID Prefix	*Metadata Object*
PF$_MR_#	Record Metadata
PF$_RS_#	Record summary metadata
PF$_TM_#	Tablemaps
PF$_DM_#	Datamaps
PF$_EX_#	Expressions
PF$_CN_#	Constraints
PF$_RL_#	Rule Metadata
PF$_DS_#	Data sets
MD$_xxx_#	Data Manager. Where xxx is the rule ID.
AB$_yyy_xxx_#	ABM. Where yyy = setID, and xxx = rule ID.

Metadata Utilities

Several utilities are available that you can use to search for and validate your metadata objects. These utilities include :

- Metadata Search engine: Enables you to search for metadata objects based on the description.
- Mass Validate: Enables you to check the validity of your metadata objects before running any PeopleSoft EPM engines or processes that depend on it.
- Impact Analysis tool: Enables you to determine the interdependencies of metadata before you change objects.

These utilities are described later in this PeopleBook.

Warning! Only an experienced user should make changes to existing metadata.

See Also

Chapter 16, "Setting Up and Working with Metadata for the Operational Warehouse - Enriched," Finding Metadata Objects, page 345

Chapter 17, "Working with Metadata Utilities," Running Mass Validate, page 407

Chapter 17, "Working with Metadata Utilities," Performing Impact Analysis, page 405

Common Elements Used in This Chapter

Hidden Object This is a check box that enables power users to edit metadata objects while ensuring that everyday users can only view the objects. It enables another level of security to be applied to metadata.

Owner ID Assign an owner ID to a particular metadata object. The owner represents an EPM functional area (such as ABM or Budgeting) that is associated with a metadata object. Assigning an owner ID to the metadata organizes and groups the metadata, making it easier to locate and audit. After an owner ID is defined for record metadata, any tablemap, datamap, and constraint built on this record inherits the owner ID of the primary table.

Note. You can add owner ID values by updating the translate values in the Owner table (PF_OWNER). However, the added values represent a configuration and are not supported and must be migrated on upgrade.

Warning! When an owner ID is changed for parent metadata (for example, record metadata), the owner IDs for child metadata (for example tablemaps and datamaps) are not updated.

SQL Object ID Prefix A system-generated number that identifies the prefix of the SQL that is built by this component in the SQL repository.

Click the Compile button on any of the metadata pages to build the metadata. Remember that you need to recompile any metadata that you change using this button.

Description The Metadata search engine uses the description that you enter to find metadata objects.

Finding Metadata Objects

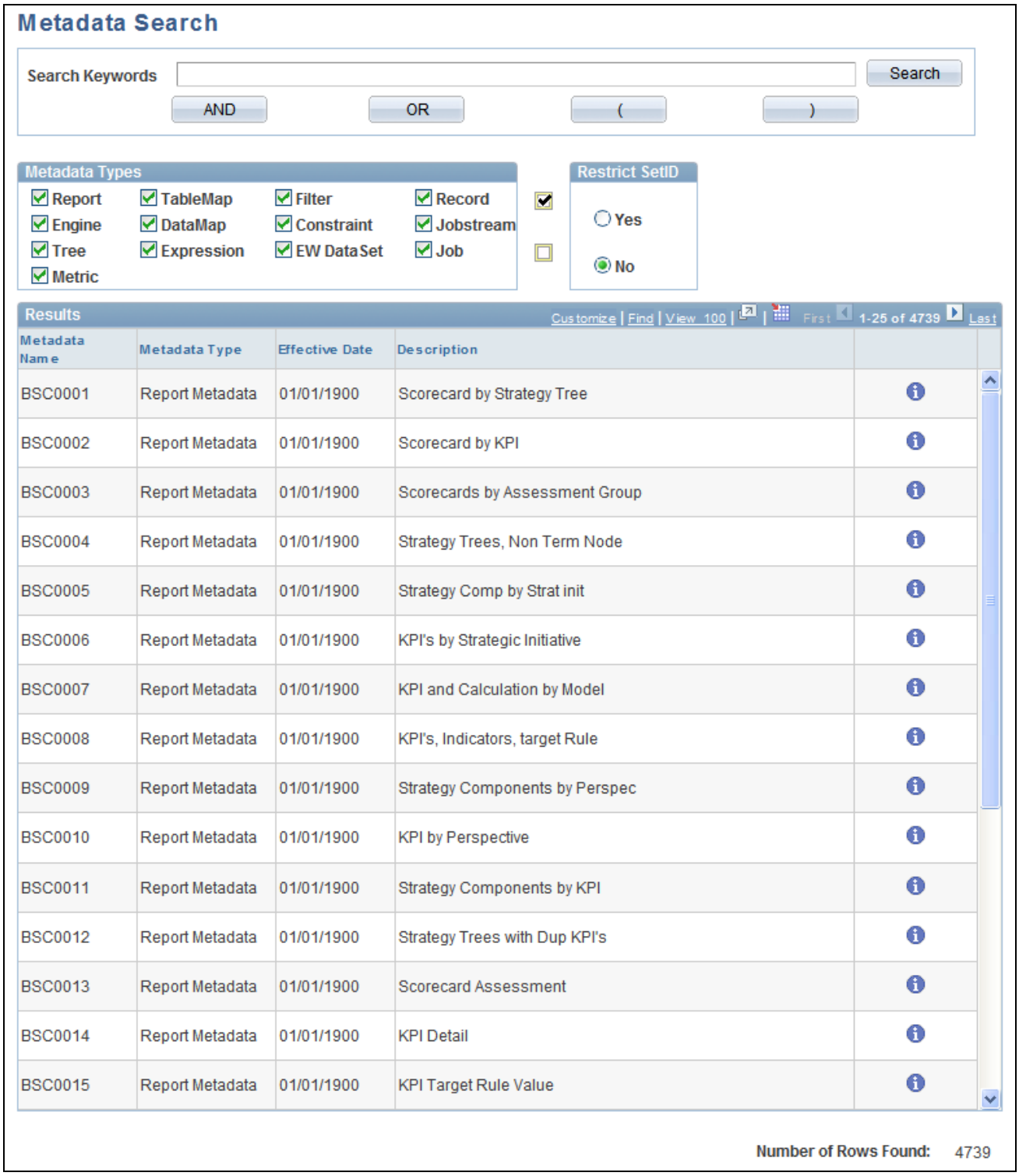
This section describes how to search for metadata objects.

Page Used to Find Metadata Objects

Page Name	*Definition Name*	*Navigation*	*Usage*
Metadata Search	PF_SRCH_PANEL	EPM Foundation, Foundation Metadata, Other Metadata Operations, Metadata Search	Search for delivered and created metadata objects.

Searching for Metadata Objects

Access the Metadata Search page (EPM Foundation, Foundation Metadata, Other Metadata Operations, Metadata Search).



Metadata Search page

Search Keywords Enter any keywords that you want to search by. You can use the AND, OR, and parentheses buttons to create a search string. Alternatively, you can leave this field blank and just search on a setID.

Metadata Types	You can further narrow your search by specifying the metadata types to search on.
	Click the All Metadata button to select all the metadata types that are listed.
	Click the Clear all flags button to clear all the current selections in the Metadata Types group box.
Restrict SetID	Select Yes to restrict the search to a specific setID. This applies only to data sets, constraints, and filters.
Search	After you have entered your criteria, click the Search button to activate the search. Depending on your criteria and the amount of metadata in your system, the search may take a while. The search returns metadata matching your criteria to the grid at the bottom of the page.
	Click the Go to Setup Page button to go directly to the metadata setup page of any of the metadata objects that are returned.

Applying the Hidden Flag to Metadata Objects

To fully secure your metadata objects, it is recommended that you establish hidden flags. Hidden flags enable power users to edit metadata objects while ensuring that everyday users can only view the objects. Hidden flags provide you with an extra level of security for your metadata. Hidden flags use the PeopleTools My Personalizations feature to determine how to render the page for a user. If the user does not have access to the hidden flag through the personalization feature, then the page will be rendered as display only. This ensures that daily users are not able to modify or delete the metadata object

Using this flag, system administrators can control access to metadata objects. System administrators may also assign power users access on an as-needed basis.

This flag acts as an attribute for the following metadata objects:

- Record metadata
- Tablemaps
- Datamaps
- Expressions
- Filters
- Constraints
- Data sets
- Dimension metadata

This section discuses how to:

- Apply the hidden flag to metadata objects.
- Grant access to hidden metadata objects.

Applying the Hidden Flag to Metadata Objects

On any of the metadata setup pages for record metadata, tablemaps, datamaps, expressions, filters, constraints, or data sets, select the Hidden Flag check box to activate the hidden flag for that metadata object.

Granting Access to Hidden Metadata Objects

System administrators may grant access to hidden metadata objects for day-to-day users (power users) by enabling them to access the My Personalizations, EPM Expert User Settings page.

On this page, users can click the Personalize Option button for EPM Expert User Settings to access the Option Category: EPM Expert User Settings page. On this page, power users can set the override value for Display Hidden Objects to *Yes* to enable them to modify metadata objects that are hidden from day-to-day users.

Setting Up Record Metadata

This section provides an overview of record metadata and discusses how to:

- Define record metadata.
- Review record column properties.
- Create and review related metadata objects.

Understanding Record Metadata

Record metadata defines the first level of metadata; that is, it defines the tables that are part of EPM . Your PeopleSoft database contains several types of tables:

- PeopleTools tables.
- PeopleSoft EPM business rules tables.
- PeopleSoft EPM data tables.

Record metadata defines and identifies the PeopleSoft EPM data tables only.

PeopleSoft delivers permanent data tables and the corresponding record metadata that identifies them as fact tables, fact reference tables, dimension tables, dimension reference tables, or transaction-dated tables.

If you add new tables, you must set up record metadata for each table that you add.

If you change a table, you must recompile record metadata for that table. If you add a non-key column to a table, you must recompile the record metadata. If you add a key column, you must recompile the record metadata and any tablemaps, datamaps, constraints, filters, or other metadata objects that are associated with it.

The Record Stub

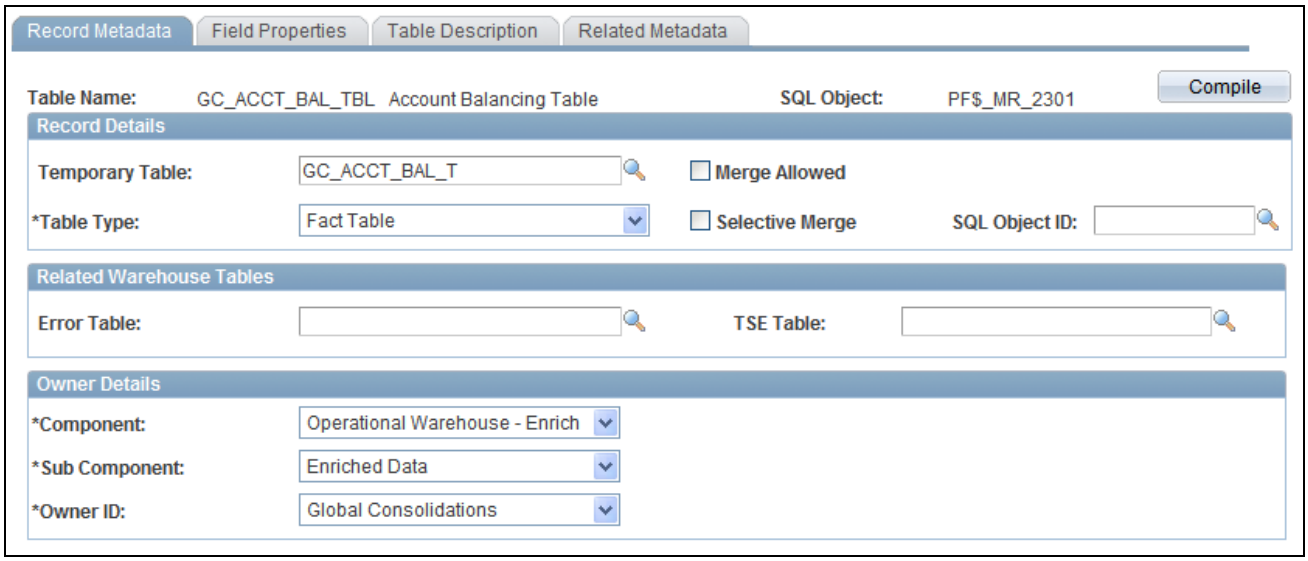
Every permanent data table that is defined within EPM requires a shadow temporary table, known as the record stub. Shadow temporary tables have a similar record layout that generally matches the permanent tables. One exception to this is long varchar, long binary columns are removed. The temporary tables are defined as temporary tables on the PeopleTools record definition.

Pages Used to Set Up Record Metadata

Page Name	*Definition Name*	*Navigation*	*Usage*
Record Metadata	PF_META_REC_TBL1	EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Record Metadata, Record Metadata	Define record metadata for any new tables that you add to EPM.
Record Metadata - Field Properties	PF_META_REC_SEQ	EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Record Metadata, Field Properties	Verify that your permanent and temporary tables are in sync.
Record Metadata - Table Description	PF_META_REC_NOTE	EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Record Metadata, Table Description	Enter any notes that you want to associate with the record metadata.
Record Metadata - Related Metadata	PF_META_RELMD	EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Record Metadata, Related Metadata	Create and review tablemaps, datamaps, and constraints that are related to specific record metadata.

Defining Record Metadata

Access the Record Metadata page (EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Record Metadata, Record Metadata).



Record Metadata page

Record Details

Temporary Table Select a temporary table to use in parallel processing.

When the system uses temporary tables for parallel processing, it uses the temporary table in conjunction with the record suite table.

Note. Temporary tables need to be selected only for Operational Warehouse - Enriched (OWE) tables.

Table Type Select the type of table for the selected record.

You can select *Dimension Reference Table, Dimension Table, Fact Reference Table, Fact Table, Security Join Table,* or *Transaction-Dated Fact Table.*

Merge Allowed Select this check box if the table name needs to be merged from a temporary table to a final table by the Merge (PF_MERGE) application engine.

Generally, this check box is used for engine output tables only.

Warning! The following tables must never be marked for Merge: PF_LEDGER_F00, PF_JRNL_F00, PF_ADB_JRNL_F00, PF_LED_ADB_F00, LEDGER, or LEDGER_ADB.

Selective Merge Select this check box to use in the selective merge delete clause.

Temporary tables that are created during the last step of a jobstream run are merged into permanent tables (F00 tables). To avoid duplicates in the permanent tables, a delete must occur before the merge. Typically, the run control parameters are used to delete the necessary rows from the permanent tables and then the temporary tables are merged in. For some isolated cases, a more restrictive delete is necessary. The Selective Merge process performs a delete with even more criteria than just the run control parameters. Selective Merge bases its delete on the run control parameters plus the SQL Object ID that is entered in the record metadata settings for each specific table and settings within the Application Engine (AE).

Note. You generally do not use Selective Merge unless your application uses KPI Manager.

SQL Object ID Select the ID of the SQL object for the selective merge.

Related Warehouse Tables

Error Table Select the OWE error table related to the selected record.

The error table contains the error data that fails as part of the edit and modification process.

TSE Table Name Select the TSE table related to the selected record.

The TSE (transaction editor set) table contains error message detail information.

Owner Details

Component Select the component to which the record metadata belongs.

Select either *Multi-Dimensional Warehouse* or *Operational Warehouse - Enrich.*

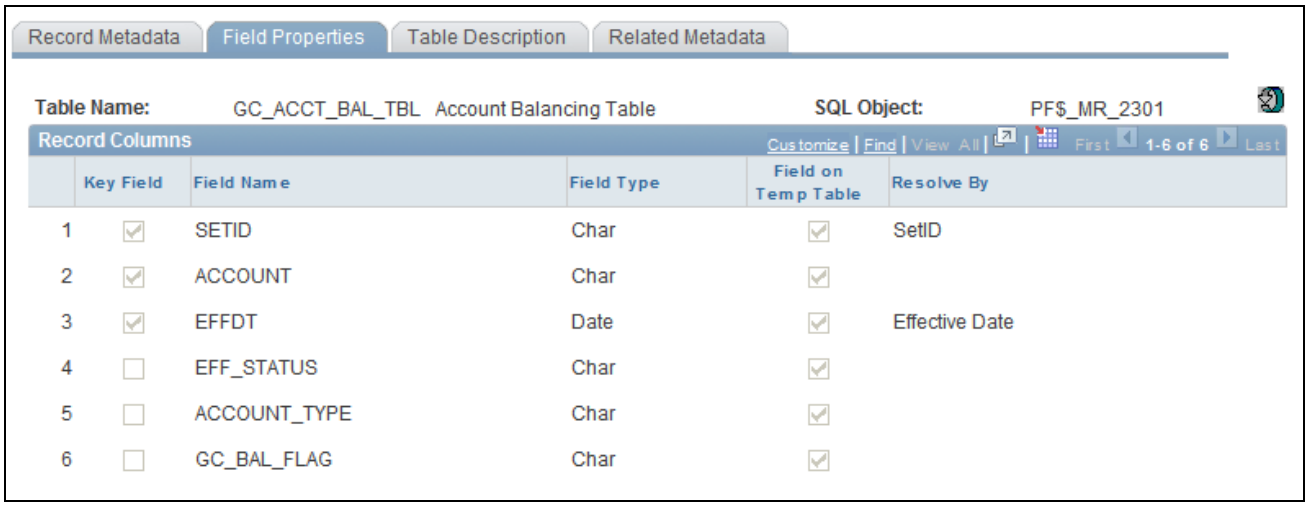
Sub Component Select the sub component, or type of data, associated with the selected record.

Select either *Enriched Data* or*Input Data.*

When you have completed your record metadata setup, click the Compile button to generate the SQL objects. You must also build your SQL anytime you make changes to the record metadata.

Reviewing Record Column Properties

Access the Field Properties page (EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Record Metadata, Field Properties).



Field Properties page

Key Field Select this check box if the field is a key on the permanent table.

Field Name Lists all the fields on the permanent table.

Field Type Lists the field type for each column.

Prompt Table Column in which you can enable the system to prompt for criteria and default value fields.

Field on Temp Table (field on temporary table) Select this check box for all fields that also appear on the temporary table. In general, this should be all fields, with the exception of transaction-dated tables.

Measure (Transaction-dated tables only). All numbers (DBFIELDTYPE = 2) and signed numbers (DBFIELDTYPE = 3) that are defined on the temporary table. When the SQL is generated for these fields in a list, they are enclosed in a sum construct, for example, sum (REPORTED_HRS).

Resolve By Select a key for use in the rule resolver process.

The Rule Resolver is an application engine program that is called by most PeopleSoft EPM engines to gather the rule sets that are used in processing. The main function of the Rule Resolver is to gather the rules for a given process run based on the standard keys of setID, business unit, model ID, fiscal year, accounting period, and effective date.

Resolver fields are used for SQL object generation that enables applications to select data based on the run control parameters.

The Resolver reduces the amount of data that an application engine needs to process by populating tables with only the data necessary for the engine to run. Individual application engines call the Resolver as part of their run process.

Note. If an OWE table is selected, the resolve by field is automatically selected and you cannot edit this field. If a MDW table is selected, you can select a resolve by field.

Click the Table Description tab to enter a more detailed description of the record metadata.

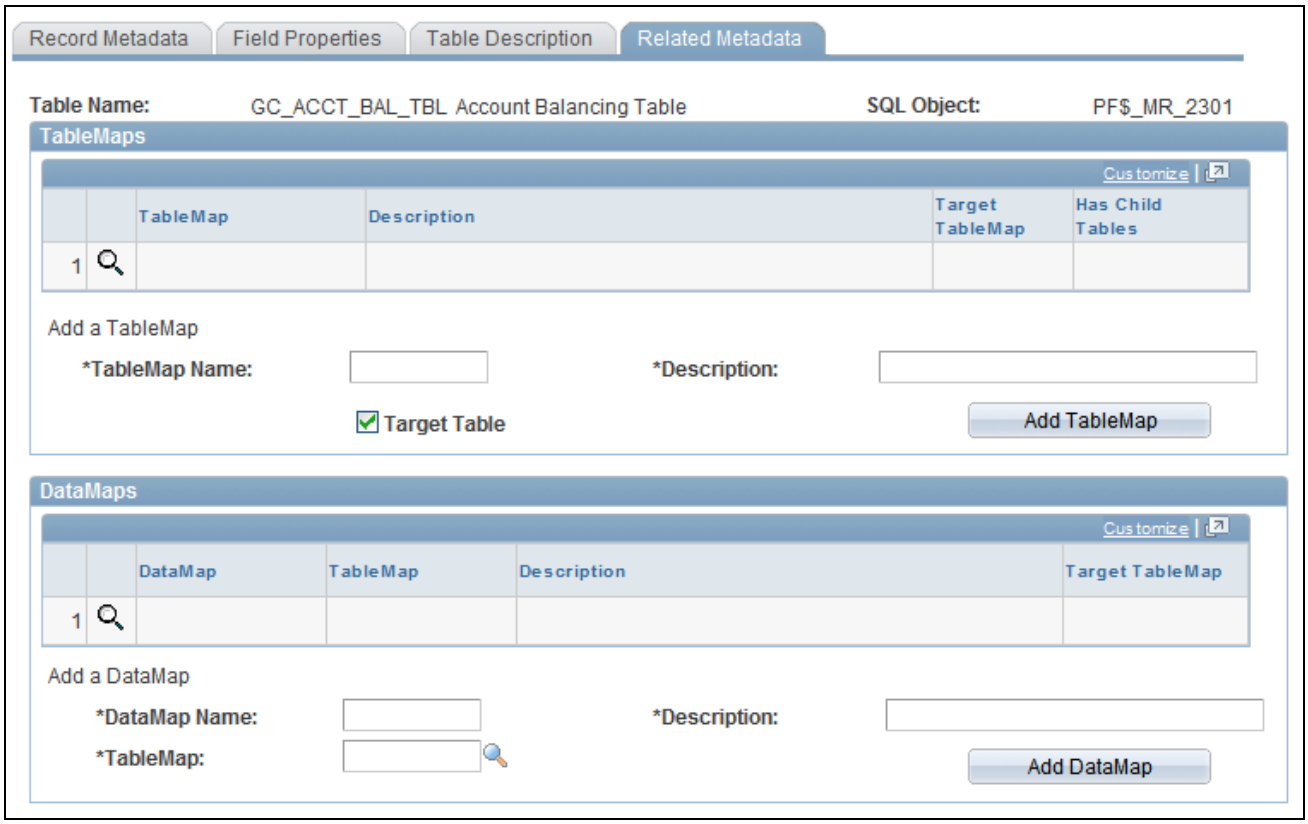
Record Columns for Transaction-Dated Tables

At resolution time, data is selected from the permanent transaction-dated fact table by transaction date and stored in the temporary table in a fiscal year and period format so that it is processed in the same way as in any other table.

For example, if an engine runs for 2002 and is based on a monthly calendar, all transaction dates between January 1, 2002 and January 31, 2002 are selected from the transaction-dated fact table and inserted into the temporary table. Because more than one transaction with the same key information may exist, some aggregation must occur. A group by clause is generated that includes all fields on the temporary table that are not resolved or are not measure fields. You use the Record Columns page to see how the fields are defined.

Creating and Reviewing Related Metadata Objects

Access the Record Metadata - Related Metadata page (EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Record Metadata, Related Metadata).



Related Metadata page

The Related Metadata page enables you to create and review tablemaps and datamaps that are related to particular record metadata. The fist time you define record metadata, you can use this page to create related tablemaps and datamaps at the same time that you create the record. Simply enter a name and description for the objects and click the add button—the objects are created automatically with the same name as the related record metadata. The SQL IDs for each object are also created automatically.

If you have already defined record metadata, you can use this page to review the tablemaps and datamaps that are built on top of the record metadata. Click any of the metadata objects to access their primary page (for example, the Tablemap page).

Setting Up Rule Metadata

This section provides an overview of rule metadata and discusses how to define rule metadata.

Understanding Rule Metadata

Rule metadata is used in conjunction with the Rule Resolver to specify the relationship between parent and child models. Unless you create a new PeopleSoft EPM engine, you will not likely need to create any rule metadata. PeopleSoft EPM engines use rule metadata to recursively determine inheritance rules between models.

Understanding the Rule Resolver

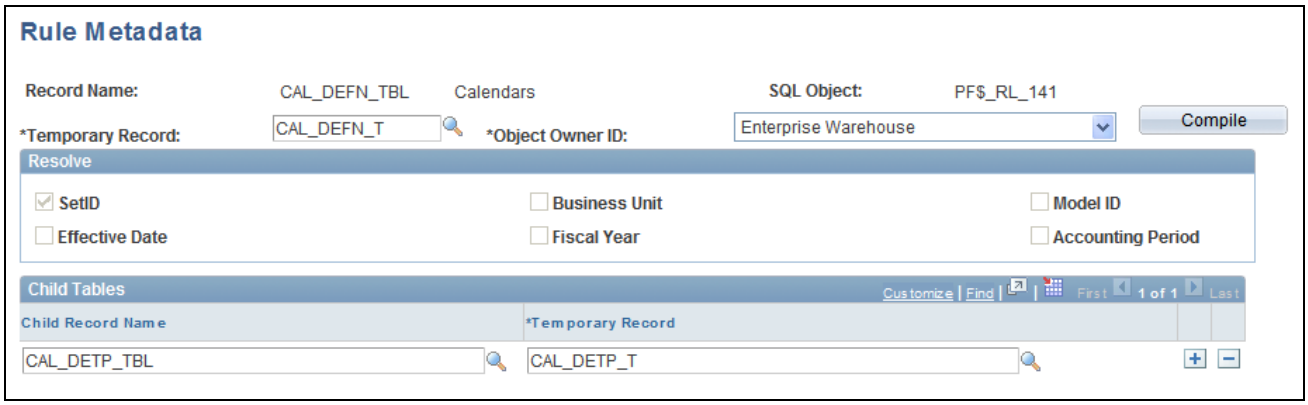
The Rule Resolver is an application engine program that is called by most PeopleSoft EPM engines to gather the rule sets that are used in processing. The main function of the Rule Resolver is to gather the rules for a given process run based on the standard keys of setID, business unit, model ID, fiscal year, accounting period, and effective date. The system does this by passing the Rule Resolver a list of parent tables that need to be resolved. The related child tables (as defined in the Rule Metadata page) are also resolved as part of the Rule Resolver. In addition, if a table is keyed by model ID, the Rule Resolver determines what the parent models are and resolves rules for them.

Page Used to Set Up Rule Metadata

Page Name	*Definition Name*	*Navigation*	*Usage*
Rule Metadata	PF_METARULE_TBL	EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Rule Metadata	Define rule metadata. Rule metadata is delivered with EPM and generates the SQL that is needed to resolve rules.

Defining Rule Metadata

Access the Rule Metadata page (EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Rule Metadata).



Rule Metadata page

Resolve The settings in this group box are set by default based on the table name that you select. You cannot update this information.

Child Record Name Select the appropriate record name. This includes all child tables that must be resolved along with the parent table.

Record Stub Select the appropriate record stub.

When you have completed your rule metadata setup, click the Compile button to generate the SQL objects. You must also build your SQL anytime you make changes to the rule metadata.

Setting Up Tablemaps

This section provides and overview of tablemaps, lists common elements, and discusses how to:

- Define tablemaps.
- Define reference tables.
- View SQL IDs.

Understanding Tablemaps

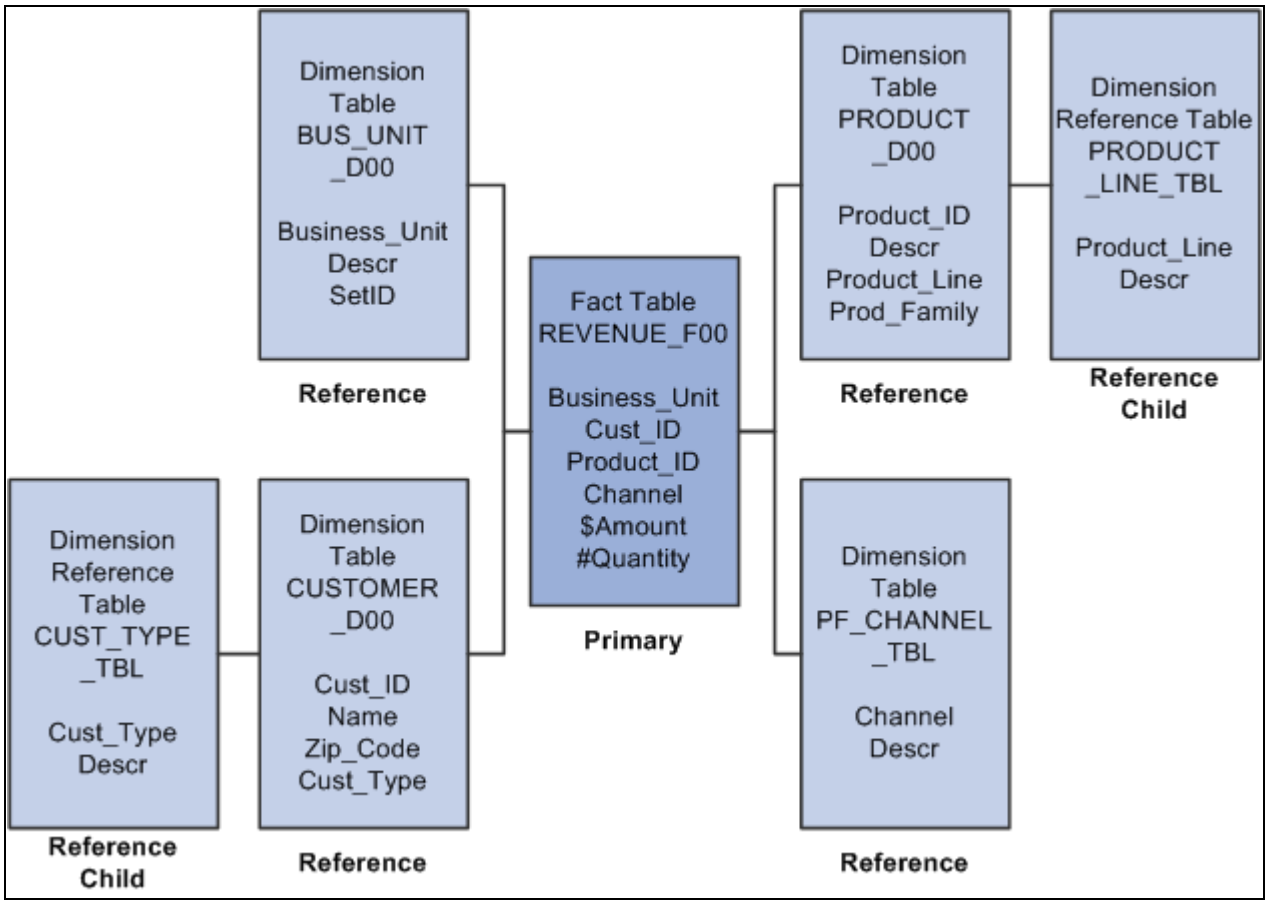
Tablemaps define the physical relationships between your PeopleSoft data warehouse tables. Tablemaps enable you to define families of related data warehouse tables and the columns that define the key relationships (or joins) between the tables. Tablemaps describe the parent-child relationships between tables by defining the common fields that join them together.

A tablemap can be defined as identifying a primary table only, or you may specify any combination of child tables, reference tables, and reference child tables:

The types of tables are:

Primary table	Often a fact table, the primary table, from a logical point of view, to which the other tables are related.
Child table	A hierarchical child of the primary table (all the same keys, plus one). Child tables enable you to have a one-to-many relationship between a primary table and its child tables and to define the relationship between these tables through the tablemaps. You specify a child table only if you need fields from that table in your tablemap.
Reference table	A table that defines the properties for one or more keys in the primary table. Reference tables contain additional attributes, or properties, of the transactional fact information. Some examples of reference tables are Customers, Products, Channels, Geography, and Time. A reference table always has one or more key fields that it shares with the primary table. These fields are defined on this page and matched with the corresponding fact field to enable engines to access data through the join. Each reference table may be the parent table to one or more reference child tables.
Reference child table	A hierarchical child of the reference table (all the same keys, plus one). Reference child tables are similar to child tables in that they allow for multiple types or rows of data per related dimension table. An example of a reference child and its related dimension is a customer demographic table's relationship to a customer table. That is, you may want to track various demographic values for each customer, such as salary range and geographic codes

The relationship between these tables is often referred to as a star schema. A star schema is a group of tables that are related to a central fact table. A single fact table can have numerous fact references and dimensions and each dimension can have numerous dimension references.



Example of relationship between tables (star schema)

Tablemaps are used as a basis for defining datamaps, filters, and ultimately constraints. After a tablemap is set up, you will likely keep it as it is. You therefore have many datamaps defined for each tablemap.

We deliver a set of tablemaps with EPM.

Warning! If you plan at the leaf level instead of the node level for a dimension, you must remove the reference table for that dimension. When you set up your data set that uses this dimension, you must select to group by dimension ID and *not* dimension node.

Common Elements Used in This Section

Join Mapping Depending on the key field type, you may or may not be able to edit this field. The system automatically resolves certain fields. Fields that you can edit are in white. If you change the method to *Map One to One =*, then you can specify the primary field. *Map One to One =* is the default if the reference key field has the same name as on the primary table.

Primary Field If you change the join mapping to *Map One to One =*, then you can specify the primary field using the valid values that are provided by the system.

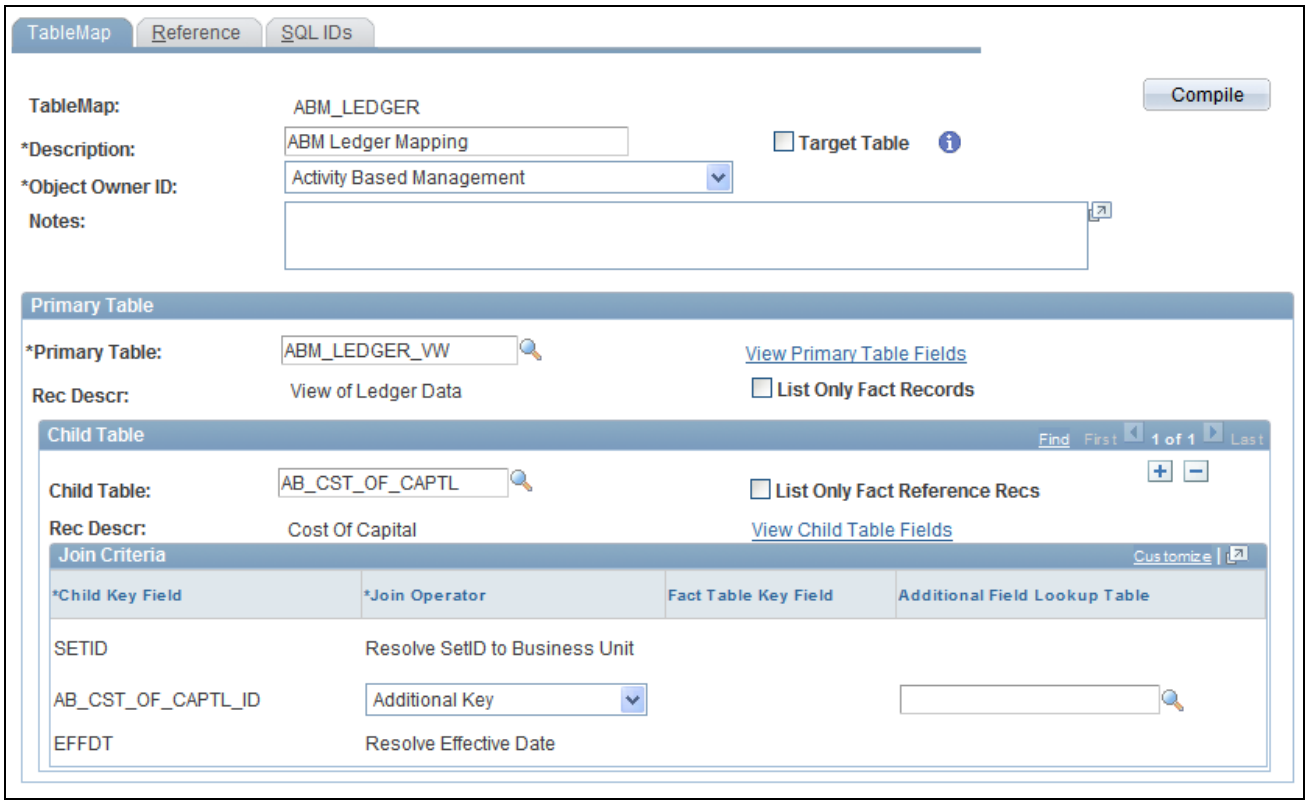
Additional Field Lookup Table If join mapping is set to *Additional Key*, you can specify that key value in this field using the valid values that are provided by the system. *Additional Key* is the default when the key cannot be resolved and if it does not have the same name on the primary table. In a valid tablemap, every child has only one additional key.

Pages Used to Set Up TableMaps

Page Name	*Definition Name*	*Navigation*	*Usage*
TableMap	PF_TBLMAP_DEFN	EPM Foundation, Foundation Metadata, Metadata Creation and Editing, TableMap, TableMap	Define tablemap and tablemap child tables.
Reference	PF_TBLMAP_REF	EPM Foundation, Foundation Metadata, Metadata Creation and Editing, TableMap, Reference	Define reference and reference child tables.
SQL IDs	PF_TBLMAP_REF	EPM Foundation, Foundation Metadata, Metadata Creation and Editing, TableMap, SQL IDs	View the generated SQL object ID prefixes for the tables that are defined in this tablemap.

Defining TableMaps

Access the TableMap page (EPM Foundation, Foundation Metadata, Metadata Creation and Editing, TableMap, TableMap).



TableMap page

Target Table

Select this check box to define the primary table as a target table for Data Manager, Allocation Manager, or another application engine.

You must select this check box if you are using this table as a target for Data Manager or Allocation. If you select this check box, the Reference page is disabled.

Note. This check box is unavailable for editing when input MDW tables are selected.

Compile

Click to generate the SQL objects for the tablemap metadata.

You must rebuild your SQL anytime you make changes to the tablemap, such as when you delete or add a key field.

Note. If you need to recompile a tablemap, you must also recompile any datamaps or other metadata objects that are associated with the tablemap.

Primary Table

Primary Table Select the table that you want to use as the primary table for this tablemap.

Only tables that have been defined in record metadata appear as valid values.

A primary table is the center of the tablemap. Generally, a primary table will be a fact table. Fact tables contain the data that is relevant to a single business transaction, and they also have a unique key structure that can identify other related fact reference or child tables. A tablemap can contain only one primary table, but it may be related to many child and reference tables.

View Primary Table Fields Click to go to the Field Properties page and review the fields that are associated with the underlying record metadata.

List Only Fact Records Select this check box if you want to view only the tables that are defined in record metadata as fact table type.

Child Fact Tables

Child Table Select the appropriate child table to join with the primary table for this tablemap.

Only tables that have been defined in record metadata appear as valid values.

List Only Fact Reference Recs(list only fact reference records) Select this check box if you want to view only the tables that are defined in record metadata as fact table type.

Child Key Field Displays the key field of the child table used to join to the fact table key field.

This key field relates the child table to the primary table.

Join Operator Select a method to join the child key field and the fact table key field.

If you select *Additional Key* to use in the join, you must specify the record in which the additional key resides.

Fact Table Key Field Select the key field of the primary fact table used to join to the child key field.

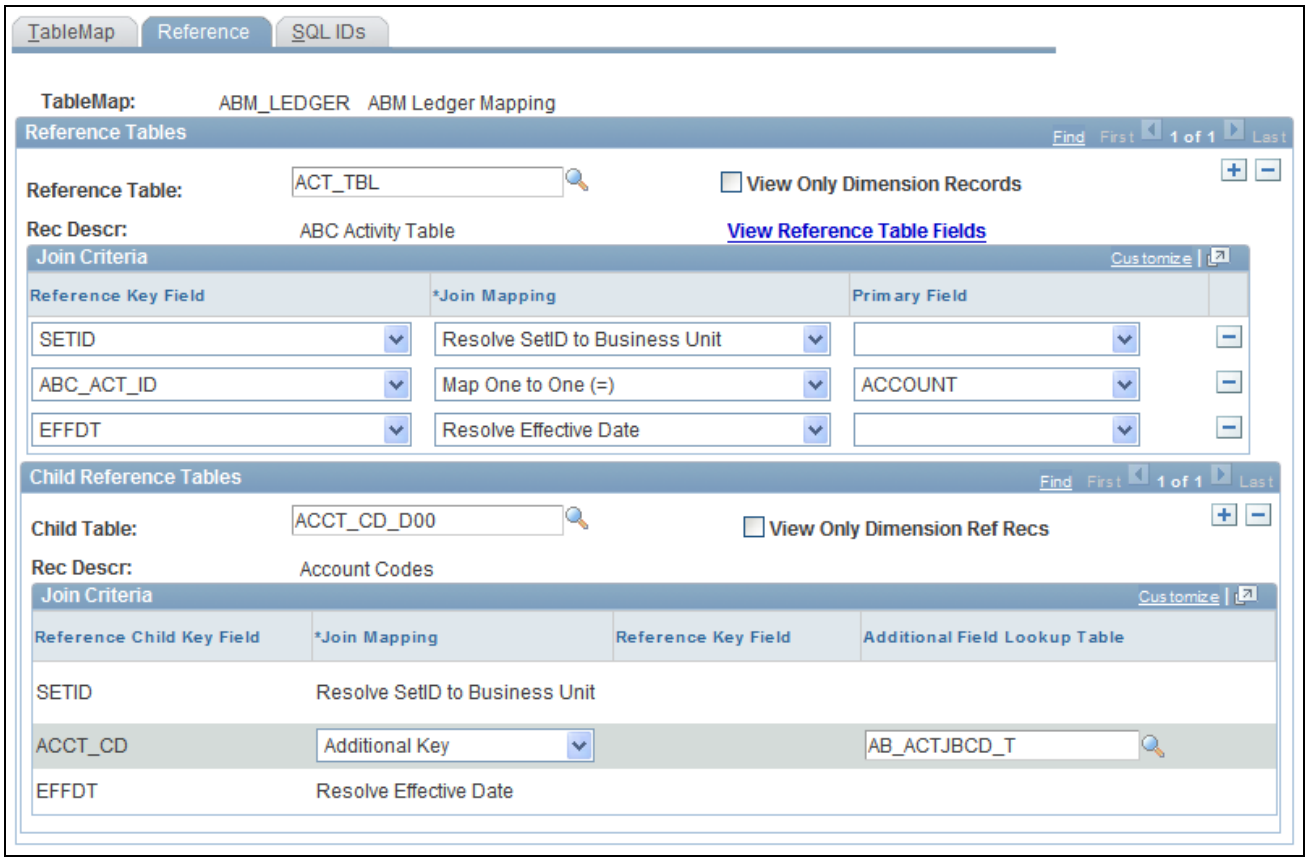
This field relates the fact table to the child table. In some cases, this field is display only.

Additional Field Lookup Table Select the record in which the additional key for the join resides.

This field is available only when you select *Additional Key* in the Join Operator field.

Defining Reference Tables

Access the TableMap - Reference page (EPM Foundation, Foundation Metadata, Metadata Creation and Editing, TableMap, Reference).



Reference page

Reference Tables

Reference Table Select the appropriate reference table that you want to join to the primary table.

Only tables that have been defined in record metadata appear as valid values.

View Only Dimension Records Select this check box to view only the tables that are defined in record metadata as dimension table type.

View Reference Table Fields Click to access the Field Properties page and view the record metadata fields related to the selected reference table.

Reference Key Field Displays the key field of the reference table used to join to the primary table key field.

Join Mapping Select a method to join the reference key field and the primary table key field.

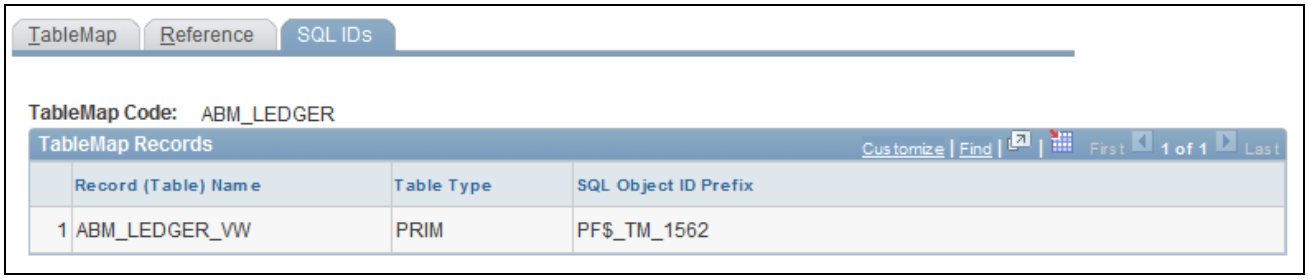
Primary Field Select the key field of the primary table used to join to the reference key field.

Child Reference Tables

Child Table	Select the appropriate reference child table that you want to join to the primary table. Only tables that have been defined in record metadata appear as valid values.
View Only Dimension Ref Recs (view only dimension reference records)	Select this check box to view only the tables that are defined in record metadata as dimension table type.
Reference Child Key Field	Displays the key field of the child reference table used to join to the reference table key field. This key field relates the child reference table to the primary reference table.
Join Mapping	Select a method to join the reference child key field and the reference table key field. If you select *Additional Key* to use in the join, you must specify the record in which the additional key resides.
Reference Key Field	Select the key field of the reference table used to join to the reference child key field.
Additional Field Lookup Table	Select the record in which the additional key for the join resides. This field is available only when you select *Additional Key* in the Join Mapping field.

Viewing SQL IDs

Access the TableMap - SQL IDs page (EPM Foundation, Foundation Metadata, Metadata Creation and Editing, TableMap, SQL IDs).



SQL IDs page

After the SQL has been compiled, you can view the generated SQL object ID prefixes for the tables that are defined in this tablemap.

Setting Up Datamaps

This section provides an overview of datamaps and discusses how to:

- Define datamaps.
- View datamap fields.
- View or add constraints for a datamap.
- Use the DataMap Wizard to create datamaps.

Understanding Datamaps

A datamap is the third level of metadata that builds upon the information that you captured in the tablemap and enables you to define a logical view of the physical EPM tables. Datamaps bring together information from the different tables that can be specified in a tablemap and defines it as if it were one entity or table. Not every column of every table that is defined in a tablemap is necessary; datamaps enable you to select only those columns that you want to use

You can group fields that are related to common processes so that processing can be done on one data set. In addition, datamaps enable you to give columns more intuitive, meaningful names, making data retrieval and review easier for business users.

Warning! If you plan at the leaf level instead of the node level for a dimension, you must remove the row that contains this dimension. When you set up your data set that uses this dimension, you must select to group by dimension ID and *not* dimension node.

Understanding Datamap Setup

PeopleSoft provides you with the option of setting up datamaps using the *DataMap Wizard*, which is intended to simplify the datamap creation process. You can still use the Record Metadata, TableMap, and DataMap components to manually set up datamaps.

See Chapter 16, "Setting Up and Working with Metadata for the Operational Warehouse - Enriched," Setting Up Record Metadata, page 348 and Chapter 16, "Setting Up and Working with Metadata for the Operational Warehouse - Enriched," Setting Up Tablemaps, page 355.

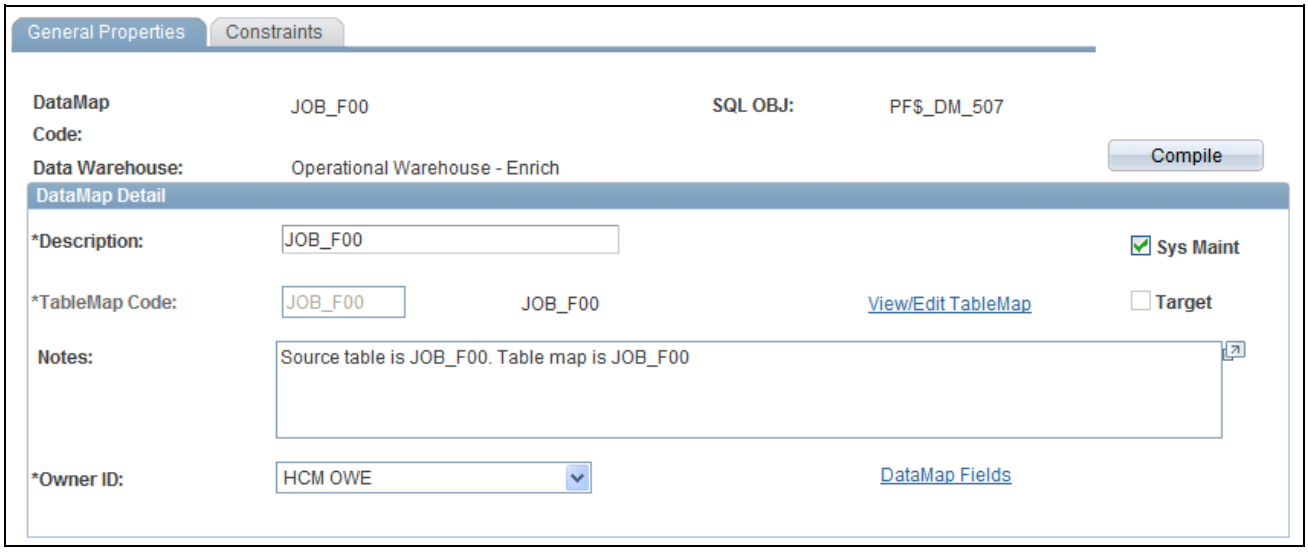
Pages Used to Set Up Datamaps

Page Name	*Definition Name*	*Navigation*	*Usage*
General Properties	PF_DATAMAP_DEFN	EPM Foundation, Foundation Metadata, Metadata Creation and Editing, DataMap, General Properties	Define datamaps.

Page Name	Definition Name	Navigation	Usage
DataMap Fields	PF_DATAMAP_DEFN2	Click DataMap Fields on the General Properties page.	View DataMap fields.
Constraints	PF_DMAP_CONSTRNTS	EPM Foundation, Foundation Metadata, Metadata Creation and Editing, DataMap, Constraints	View or add constraints for a specific datamap.
Datamap Wizard	PF_DATAMAP_WIZ, PF_DATAMAP_SUM	EPM Foundation, Foundation Metadata, Metadata Wizards, Datamap Wizard	Simplify datamap setup using the Datamap Wizard.

Defining Datamaps

Access the DataMap - General Properties page (EPM Foundation, Foundation Metadata, Metadata Creation and Editing, DataMap, General Properties).



DataMap - General Properties page

TableMap Code Select a tablemap code. The code selected here limits the records that you can include in your datamap to the ones that are defined in the tablemap. When you add or change this value, the page automatically inserts each field from the tablemap's primary table into the grid. Generally, you include all fields from the primary table and add only a few from the reference tables. You can add or remove rows from the grid below. If your tablemap is a target table, you cannot insert or delete rows.

View/Edit TableMap Click to access the TableMap page and view the tablemap associated with the selected datamap.

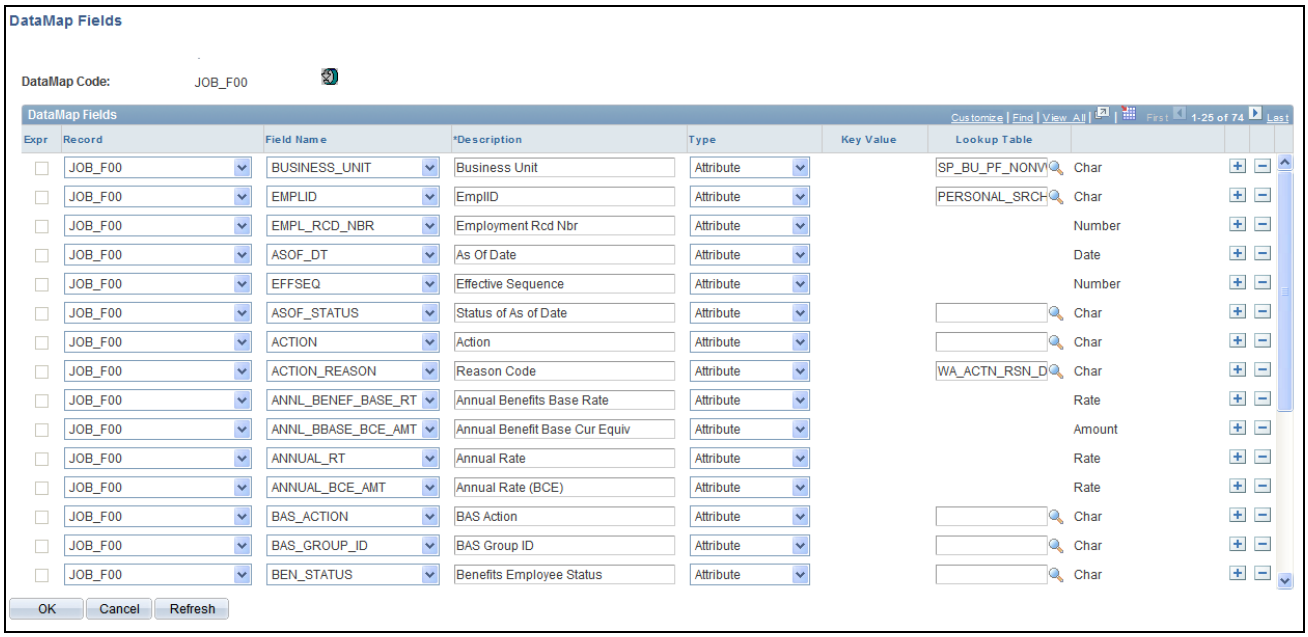
Target	This check box is controlled by the tablemap that the datamap is based on. If the tablemap that you select is a target, the check box is selected.
DataMap Fields	Click to access the DataMap - DataMap Fields page and view or edit the fields that are associated with this datamap.

When you have completed your datamap setup, click the Compile button to generate the SQL objects. You must also build your SQL anytime you make changes to the datamap, such as when you delete or add a key field.

Note. If you need to recompile a datamap, you must also recompile any tablemaps or other metadata objects that are associated with the datamap.

Viewing DataMap Fields

Access the DataMap - DataMap Fields page (Click DataMap Fields link on the General Properties page).



DataMap Fields page

Expr (expression) To create a virtual column, add a new blank row, select the Expr check box, and click the Expression button. This accesses the Expressions page.

After you have defined and saved your expression, all the changes made to the DataMap page are also saved. You cannot change an existing row into an expression.

See Chapter 16, "Setting Up and Working with Metadata for the Operational Warehouse - Enriched," Setting Up Expressions, page 373.

Record Contains a list of all the records that are defined in the tablemap on which this datamap is based.

Field Name Lists all the possible fields for the selected record. When you add or change a field, the description is automatically populated with the column field name that you select. You may then edit the description to be anything you want under 30 characters long.

Description Change the name of a field to enable you to create more intuitive names that provide more meaning than the original column name when creating your datamap. The names that you select here will show up on reports and are used in filters, constraints, and by the Data Manager.

Type Select *Attribute*, *Dimension*, or *Measure*. The system's default is *Attribute*.

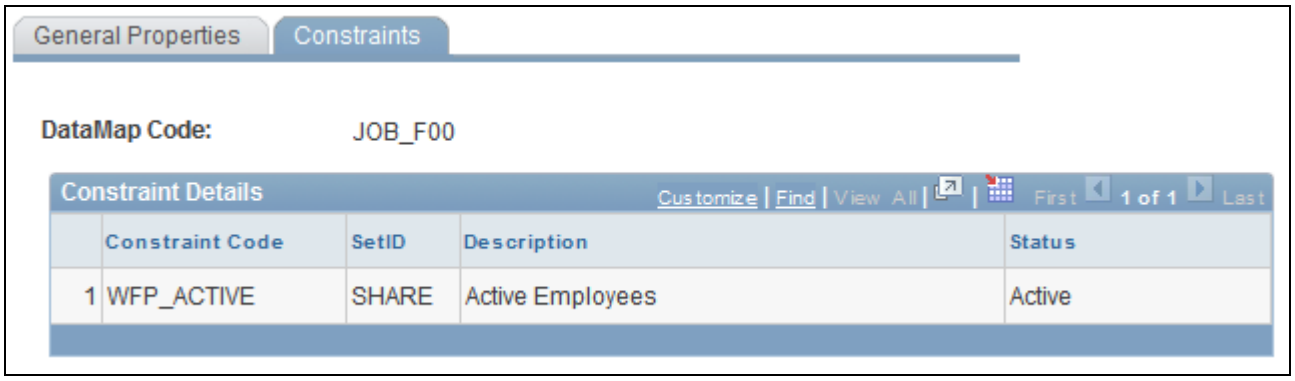
The type field is important when you are defining datamaps for the Data Manager to verify rule definitions.

Note. The types mentioned here are different from the types that are used by PeopleSoft EPM metadata and OLAP objects.

Key Value	Required for fields from child tables. Enables you to define more than one column from a single column based on different values in the lookup table code on the child table.
Lookup Table	This field is available for character fields and is used when you are defining filters to limit the field to valid values that are contained in the Lookup table.
Field Type	Displays the field type that is associated with the field. Its value is based on the field type that is defined in PSDBFIELD. Number fields consist of the following types: 1 (Basis Points), 2 (Rate), 3 (Amount), and 5 (Number). These types represent ways in which the field will be summed.

Viewing DataMap Constraints

Access the Constraints page (EPM Foundation, Foundation Metadata, Metadata Creation and Editing, DataMap, Constraints).



DataMap - Constraints page

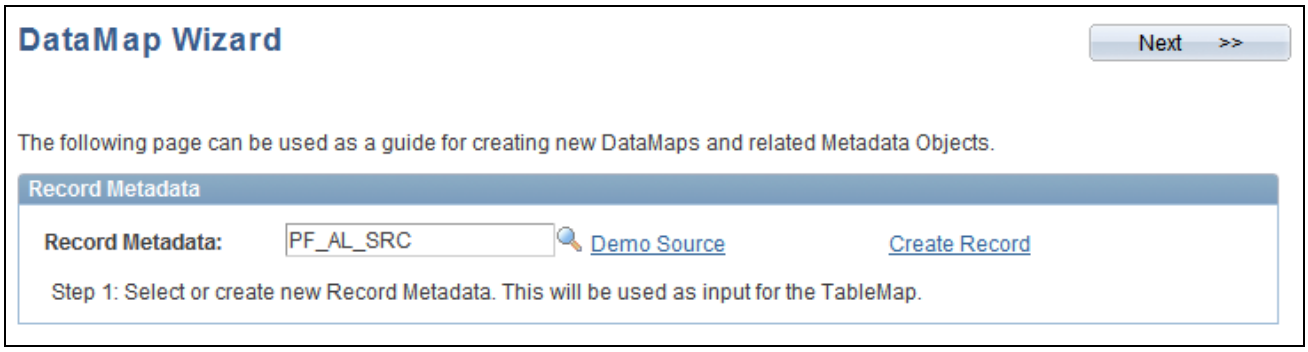
You can use this page to view constraints associated with a datamap. You can also add new constraints to the datamap.

Setting Up Datamaps using the Datamap Wizard

Access the Datamap Wizard (EPM Foundation, Foundation Metadata, Metadata Wizards, Datamap Wizard)

DataMap Wizard - Record Metadata (Step 1 of 6)

The first view of the DataMap Wizard enables you to define Record metadata for your DataMap.



Datamap Wizard - Record Metadata input

Next Click to advance to the next step of the DataMap Wizard and confirm your record metadata selection.

Record Metadata Select the record metadata (primary table) to associate with the tablemap (which is used as the basis for your datamap)

Generally, a primary table will be a fact table.

Only tables that have been defined in record metadata appear as valid values.

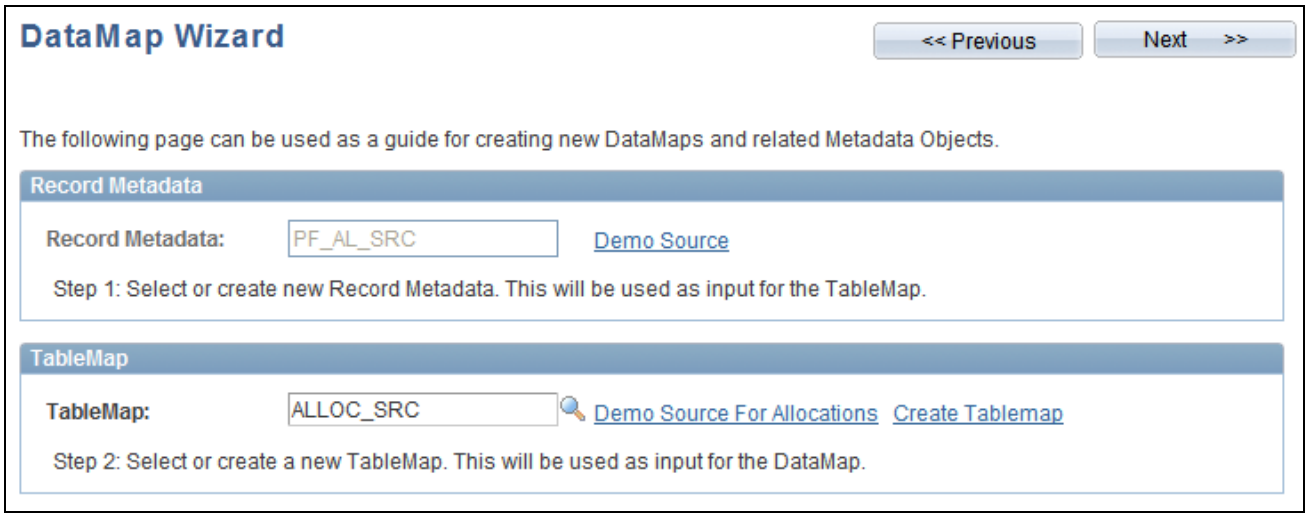
Create Record Click to access the Record Metadata page and create a new primary table to associate with the tablemap.

Note. Once you select record metadata for the wizard, a new link appears that can access the Record Metadata page for that specific record. The link name is determined by the description provided on the Record Metadata page.

In the example above the link is named *Demo Source* because that is the description given for the PF_AL_SRC record on the Record Metadata page.

DataMap Wizard - TableMap Metadata (Step 2 of 6)

The second view of the DataMap Wizard enables you to define TableMap metadata for your DataMap.



Datamap Wizard - TableMap Metadata input

Previous Click to return to the previous step of the DataMap Wizard.

Next Click to advance to the next step of the DataMap Wizard and confirm your tablemap selection.

TableMap Select a tablemap to associate with the datamap.

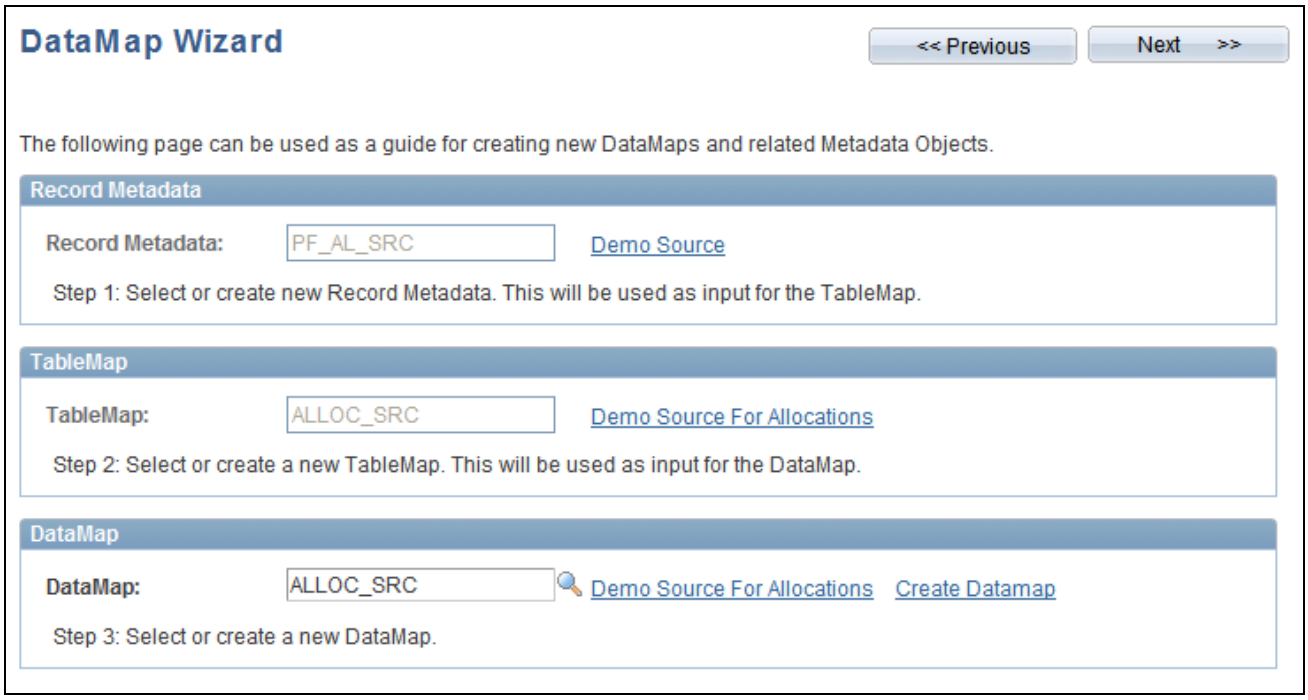
Create TableMap Click to access the TableMap page and create a new tablemap to associate with the datamap.

Once you select a tablemap for the wizard, a new link appears that can access the TableMap page for that specific tablemap. The link name is determined by the description provided on the TableMap page.

Note that as you progress through the DataMap Wizard steps, the previous section becomes unavailable for input. You can click Previous at any time to return to the previous step.

DataMap Wizard - DataMap Metadata (Step 3 of 6)

The third view of the DataMap Wizard enables you to define your DataMap.



Datamap Wizard - DataMap Metadata input

Previous Click to return to the previous step of the DataMap Wizard.

Next Click to advance to the next step of the DataMap Wizard and confirm your datamap selection.

DataMap Select a datamap.

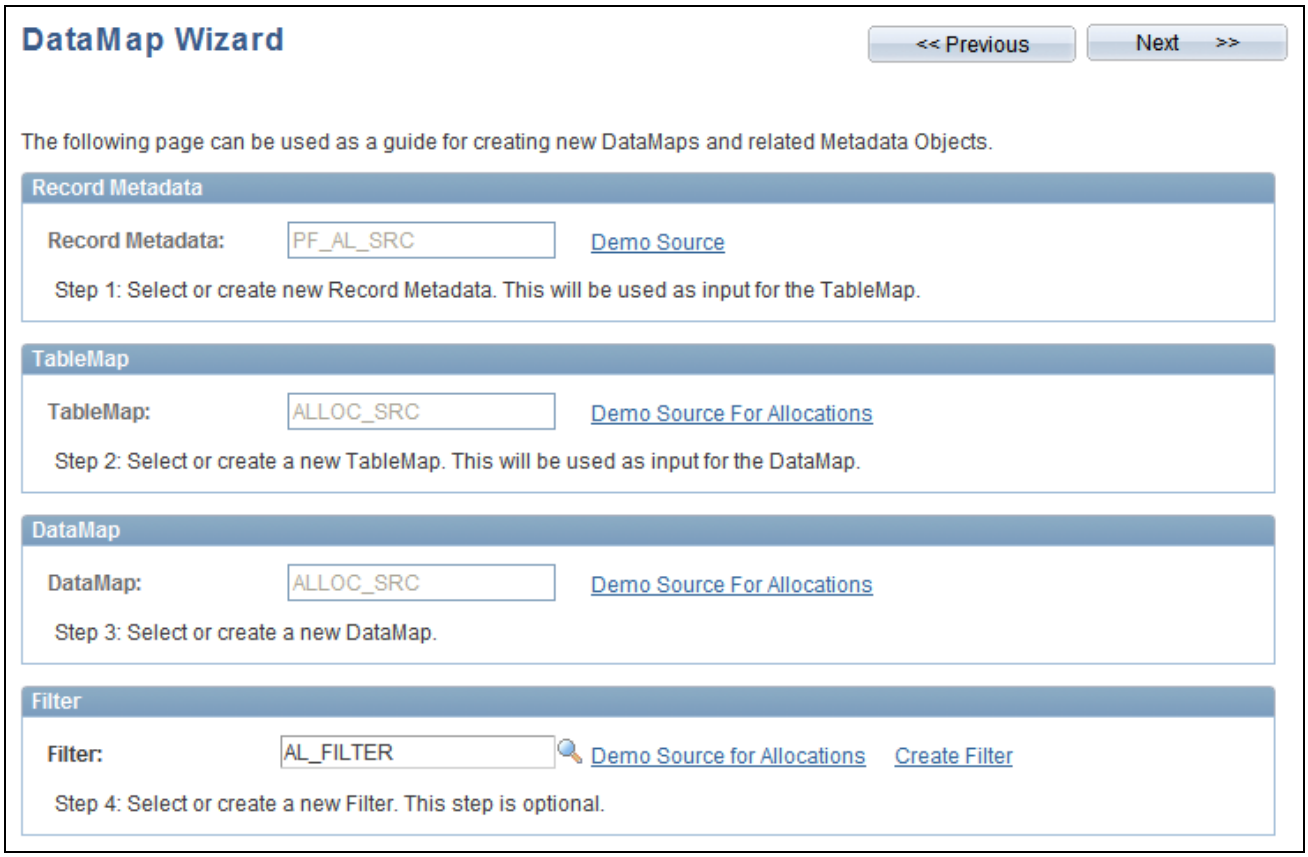
Create DataMap Click to access the DataMap page and create a new datamap.

Once you select a datamap for the wizard, a new link appears that can access the DataMap page for that specific datamap. The link name is determined by the description provided on the DataMap page.

Note that as you progress through the DataMap Wizard steps, the previous section becomes unavailable for input. You can click Previous at any time to return to the previous step.

DataMap Wizard - Filter Metadata (Step 4 of 6)

The fourth view of the DataMap Wizard enables you to define filter metadata for your DataMap.



Datamap Wizard - Filter Metadata input

Previous Click to return to the previous step of the DataMap Wizard.

Next Click to advance to the next step of the DataMap Wizard and confirm your filter selection.

Filter Select a filter to associate with the datamap.

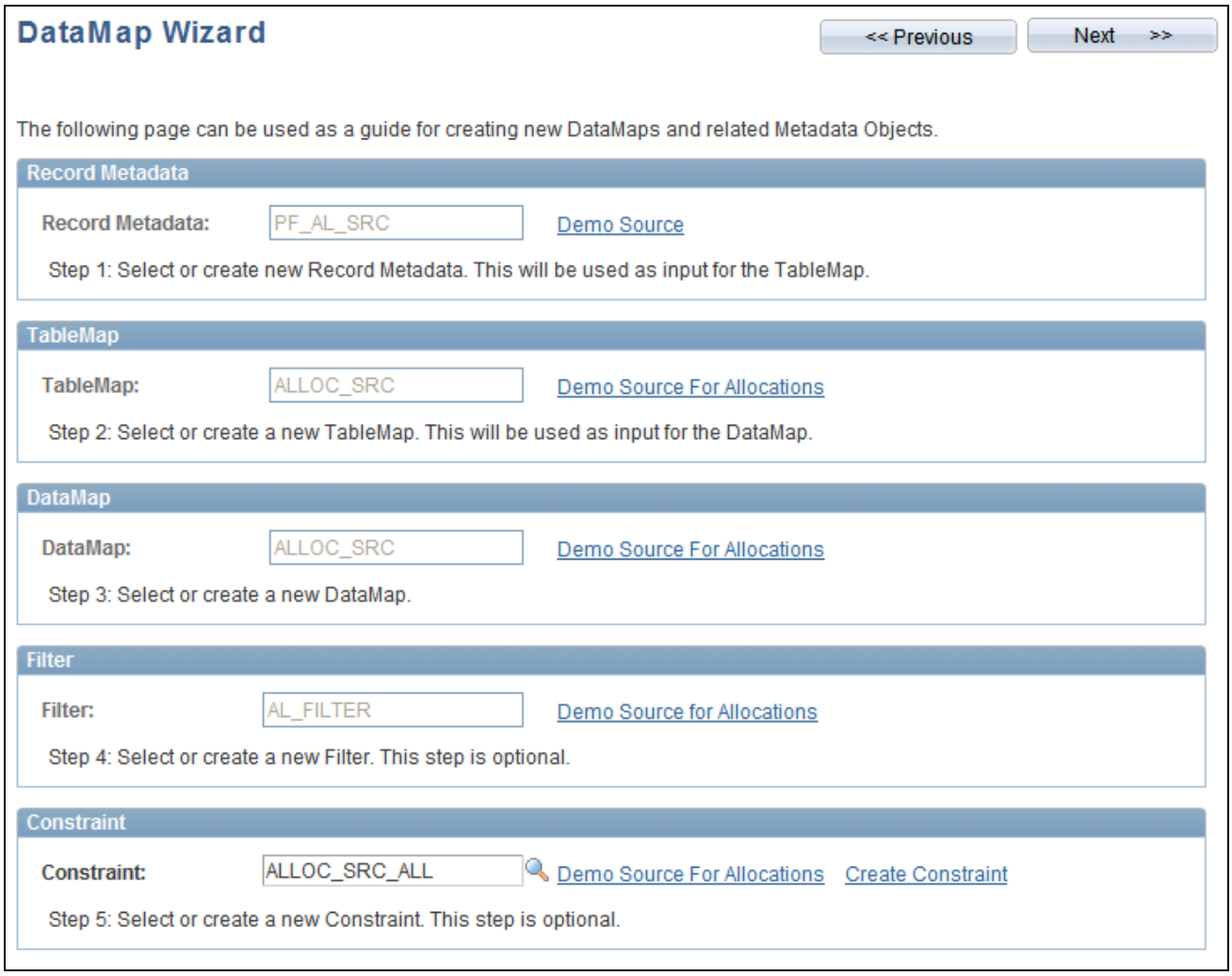
Create Filter Click to access the Filter page and create a new filter.

Once you select a filter for the wizard, a new link appears that can access the Filter page for that specific filter. The link name is determined by the description provided on the Filter page.

Note that as you progress through the DataMap Wizard steps, the previous section becomes unavailable for input. You can click Previous at any time to return to the previous step.

DataMap Wizard - Constraint Metadata (Step 5 of 6)

The fifth view of the DataMap Wizard enables you to define constraint metadata for your DataMap.



Datamap Wizard - Constraint Metadata input

Previous Click to return to the previous step of the DataMap Wizard.

Next Click to advance to the next step of the DataMap Wizard and confirm your constraint selection.

Constraint Select a constraint to associate with the datamap.

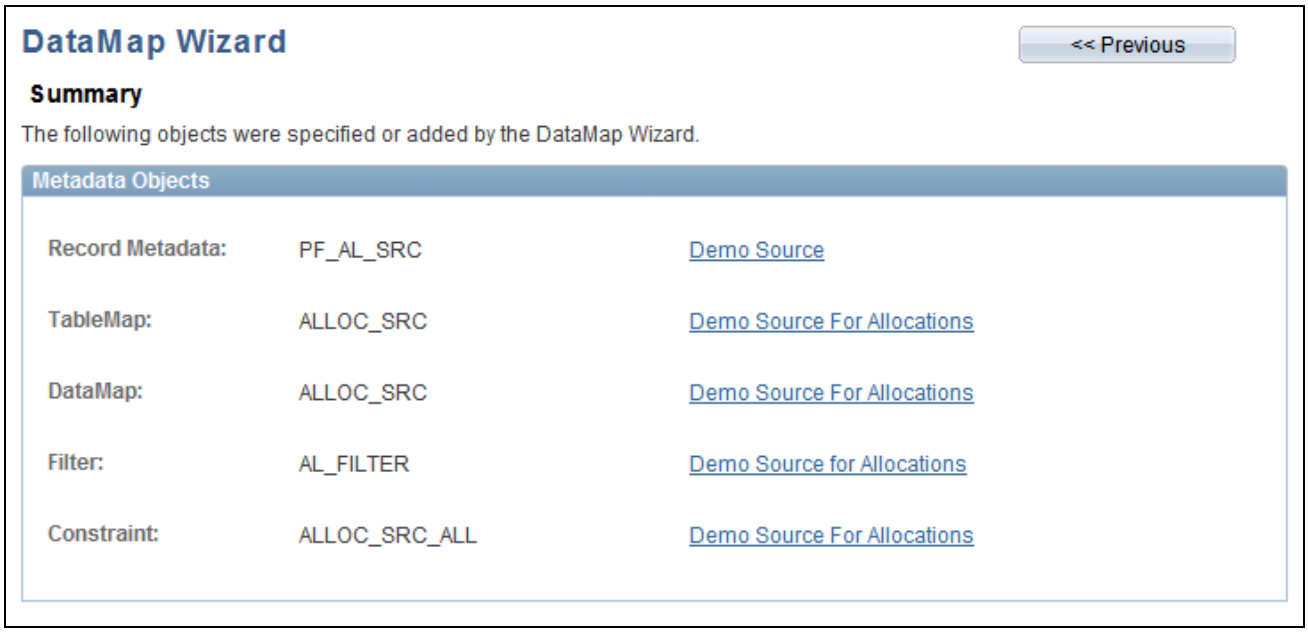
Create Constraint Click to access the Constraint page and create a new constraint.

Once you select a constraint for the wizard, a new link appears that can access the Constraint page for that specific constraint. The link name is determined by the description provided on the Constraint page.

Note that as you progress through the DataMap Wizard steps, the previous section becomes unavailable for input. You can click Previous at any time to return to the previous step.

DataMap Wizard - DataMap Summary (Step 6 of 6)

The sixth and final view of the DataMap Wizard provides you with a summary of all the metadata defined for your DataMap.



Datamap Wizard - DataMap summary

You can click on the links to the right of each metadata object to return to their main criteria pages and make adjustments to the rules.

You can also click Previous to return to the previous DataMap Wizard steps.

Setting Up Expressions

Expressions enable you to create virtual columns that are made up of mathematical calculations based on actual fields on a table. Because expressions are resolved at runtime, duplicate information is not stored in the database. Expressions are user-defined columns that you add to a datamap. After you add them as columns to the underlying datamap, you can use them in filters just like record fields. An expression can be either numeric or a string. Numeric expressions can combine any number of record fields, value objects, and math operators. String expressions can be concatenations of any number of character record fields and value objects.

Note. On DB2 UDB for OS/390 and z/OS, you should compose your expressions so that multiplication occurs before division; otherwise, decimal precision may be affected as values may be truncated. Use parentheses where necessary to control the order of calculation to ensure correct decimal precision.

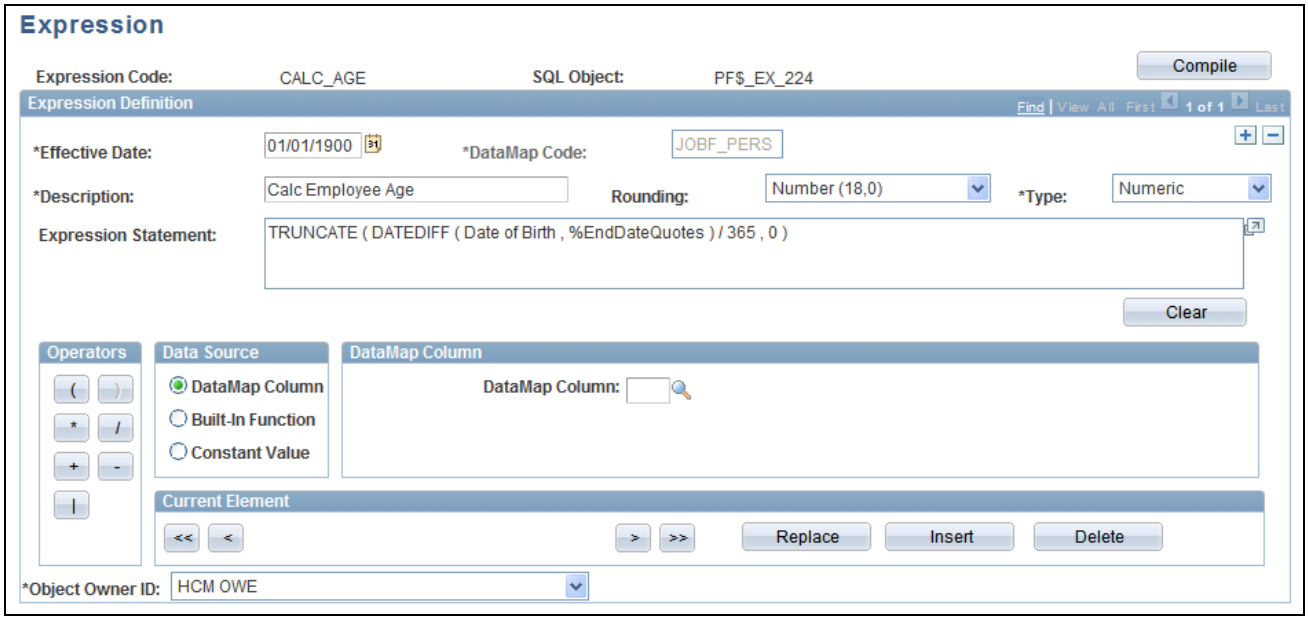
This section discusses how to define expressions.

Page Used to Set Up Expressions

Page Name	Definition Name	Navigation	Usage
Expression	PF_EXPR_DEFN	EPM Foundation, Business Metadata, Constraint and Expressions, Expression	Define an expression.

Defining Expressions

Access the Expression page (EPM Foundation, Business Metadata, Constraint and Expressions, Expression).



Expression page

DataMap Code Select the datamap that you want to build your expression on. You cannot change the datamap code after you have associated a datamap with an expression.

Rounding For numeric expressions, the result of the expression is rounded based on the precision that is selected. For all the other types of expressions, the rounding factor is not applicable.

Type Controls the set of operators, record fields, and value objects that can be selected. Values are:

Numeric: These expressions can combine any number of record fields, value objects, and math operators.

String: These expressions can be concatenations of any number of character record fields and value objects.

Date: These expressions can be a constant data or a date field that is derived from the datamap or built-in functions.

Expression Statement Displays the expression that you build by selecting operators, fields, or value objects in the group boxes below. This is where you build your expression logic. You cannot type directly in the Expression Statement field.

First, select a data source. The data source that you select presents you with different options below it. To add an element to the expression statement, select it from the data source, and then click the Insert button. The system inserts your selection into the text area. Use the arrow buttons at the bottom to move in the text area. You can see what element is being specified by the arrows surrounding it, for example >>*Effective Date*<<.

Note. >>*Effective Date*<< is also displayed in the Current Element group box.

Operators Select from the operators buttons to add the indicated value to the expression.

DataMap Column Select to display a datamap column drop-down list box. Select the appropriate datamap column from the list and use the Insert button to place your selection in the text area of the page.

Built-In Function Select to display a built-in function drop-down list box. Select the appropriate function from the list and use the Insert button to place your selection in the text area of the page. Built-in functions enable you to define numeric calculations, for example sum, average, or end of month, and minimum and maximum values.

Constant Value Select to make available the constants options and, depending on your choice, a drop-down list box. The Ad Hoc option enables you to enter your own values for the character, numeric, and date field. The From List drop-down list box contains valid value objects. Use the Insert button to place your selection in the text area of the page.

Note. It is important to consider how an expression will be used when you want to specify date-related value objects. If the expression is to be used in a WHERE clause, then the %CurrentDateIn value object must be used. If the expression is to be used in a SELECT clause, then the %CurrentDateOut value object must be used.

Clear Click to erase all of the text in the Expression Statement box.

First, Previous, Next, Last	Use these arrow buttons to move in the Expression Statement area of the page and identify the text that you want to replace, insert, or delete. When you click either > or < , you move one element forward or backwards. If you use the double arrows >> or <<, you move either to the beginning of the expression text or to the end. You can see what element is being specified by the arrows surrounding it, for example >>*Effective Date*<<.
Replace	Use the arrow buttons to specify the element that you want to replace, select another element, and then click Replace.
Insert	After you have selected a data source, use this button to insert it into the Expression Statement text area. The system inserts the element at the position designated by the work area, pushing everything else to the back.
Delete	Use the arrow buttons to specify the element that you want to delete, and then click Delete.

When you have completed your expression, click the Compile button to compile the expression.

SQL Functions That Are Available to the Expression Builder

The following table lists the SQL functions that are available to build expressions.

SQL Object ID	*Expression Function*	*Return Value*	*Description*
PF_FUNCLIB_DATE_DAYADD_UPD	AddtoDays (Date, Integer)	Date	Increase date by adding days (Integer).
PF_FUNCLIB_DATE_DAYDIFF_UPD	DiffDates (Date, Date)	Integer	Calculate difference between two dates.
PF_FUNCLIB_DATE_DAY_UPD	GetDay(Date)	Integer	Returns numeric day of the month from date.
PF_FUNCLIB_DATE_MONADD_UPD	AddtoMonth(Date, Integer)	Date	Increase date by adding months (Integer).
PF_FUNCLIB_DATE_MONBEG_UPD	BOM(Date)	Date	Returns the date value for the beginning of the month.
PF_FUNCLIB_DATE_MONTH_UPD	GetMonth(Date)	Integer	Returns numeric month from date.
PF_FUNCLIB_DATE_YEARBEG_UPD	BOY(Date)	Date	Returns date for the beginning of the year.

SQL Object ID	Expression Function	Return Value	Description
PF_FUNCLIB_DATE_YEAREND_UPD	EOY(Date)	Date	Returns date for the end of the year.
PF_FUNCLIB_DATE_YEAR_UPD	GetYear(Date)	Integer	Returns numeric year from date.
PF_FUNCLIB_DATE_YYMMDD_UPD	YYMMDD(Date)	Character	Formats date YYMMDD.
PF_FUNCLIB_DIFF_H(DTTM,DTTM)	DTTMDIFF_H(Date, Date)	Integer	Date time difference in hours.
PF_FUNCLIB_DTTM_DTTMDIFF_M_UPD	DTTMDIFF_M(Date, Date)	Integer	Date time difference in minutes.
PF_FUNCLIB_DIFF_S(DTTM,DTTM)	DTTMDIFF_S(Date, Date)	Integer	Date time difference in seconds.
PF_FUNCLIB_MATH_ABS_UPD	ABS(Integer)	Integer	Absolute value.
PF_FUNCLIB_MATH_MOD_UPD	MOD(Integer, Integer)	Integer	Modulus.
PF_FUNCLIB_MATH_TO_NUMB_UPD	TO_NUM(Character)	Integer	Convert to number.
PF_FUNCLIB_RTRIM	RTRIM(Character)	Character	RTrim blanks.
PF_FUNCLIB_TO_CHAR_UPD	TO_CHAR(Integer)	Character	Convert to character.

Note. All expression functions are operating system independent and database dependent.

Using Data Sets

Data sets are used as input for various engines, for instance, the Forecasting engine, user-defined functions, drivers in PeopleSoft Activity-Based Management (PeopleSoft ABM), and data elements in PeopleSoft KPI Manager. Data sets provide a user-defined set of information to the engines. Data sets restrict used columns and returned rows using constraints.

When you create a data element in PeopleSoft KPI Manager, for example, you are actually creating a data set. Though each data set is created by a process-specific setup, the underlying logic is the same, enabling you to more easily understand the functional aspects of the process.

Note. You will most likely not need to create a data set using the DataSet page because data sets are created behind the scenes. However, you may have to rebuild or recompile a data set if you change an underlying table. Data sets that are created in PeopleSoft EPM analytical applications do not appear on this page.

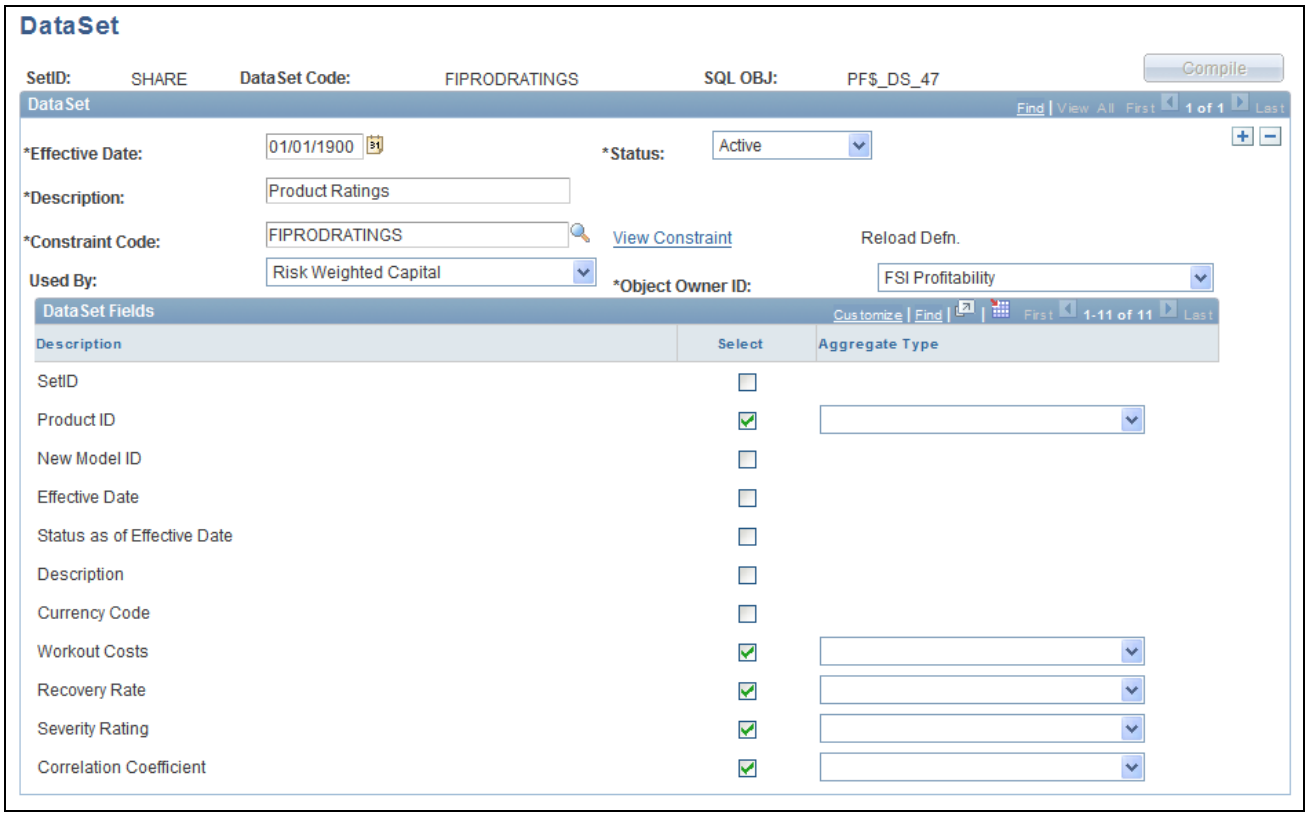
This section provides an overview of data sets and discusses how to recompile or rebuild data sets.

Page Used to Use Data Sets

Page Name	*Definition Name*	*Navigation*	*Usage*
DataSet	PF_DATASET_DEFN	EPM Foundation, Foundation Metadata, Metadata Creation and Editing, DataSet	Recompile or rebuild a data set.

Recompiling or Rebuilding DataSets

Access the DataSet page (EPM Foundation, Foundation Metadata, Metadata Creation and Editing, DataSet).



DataSet page

Constraint Code The constraint code for this data set. The selected constraint code populates the grid at the bottom of the page.

Select Select this check box to indicate that this datamap column is included in the data set's select clause.

Aggregate Type The aggregate type choices correspond to the standard SQL aggregate functions. If you select an aggregate type, all the other selected columns are populated with the group by default. Values are:

Avg (average): Returns the average of all the records retrieved.

Avg Distinct: Returns the average of all the unique records retrieved. For example, average (5, 8, 9, 12, 9, 7, 5) = 55/7 but average distinct (5, 8, 9, 12, 9, 7, 5) = 41/5.

Count: Returns the count of all the records retrieved.

Count Distinct: Returns the count of all the unique records retrieved. For example, count (A, B, A, C) = 4 but count distinct (A, B, A, C) = 3.

Group By: Groups the records retrieved by the measure field.

Min (minimum): Returns the minimum of all the records retrieved.

Max (maximum): Returns the maximum of all the records retrieved.

Sum: Returns the sum of all the records retrieved.

Sum Distinct: Returns the sum of all the unique records retrieved. For example, sum (1,1) = 2 but sum distinct (1,1) = 1.

Select the Compile button to compile the data set. This compilation sets the as of date of the data set to the effective date. If you want to change the as of date, you must recompile after updating the Effective Date field.

Use the Mass Validate utility to validate against as of dates.

Note. If a data set is sent to Resolver and has an inactive constraint, the system returns an error message.

PF_FETCH

PF_FETCH is an application engine process that is used by PeopleSoft EPM analytical applications to retrieve and display data by running the SQL behind the data set for a set of run control parameters after the data set is built (SQL is generated). PF_FETCH selects and inserts data into an output table that is specified by the application. Output table data is automatically deleted if it is more than two days old based on the date and time stamp.

Setting Up Filters

This section provides an overview of filters and discusses how to:

- Define filters.
- Specify filter selection criteria.
- Use the tree viewer.

Understanding Filters

Filters are used extensively by the PeopleSoft EPM engines to define subsets of data to perform operations or calculations on. In your datamap, you define which columns from the tablemap to use in processing. Filters enable you to specify which rows to use from those columns, similar to a WHERE clause in a query.

In general, the PeopleSoft EPM engines use set-based processing to process large amounts of data as efficiently as possible. Filters enable you to define what subset of data gets processed by or uses a specific business rule. Not every row of data may be necessary to process your data. Filters enable you to select only those rows you want.

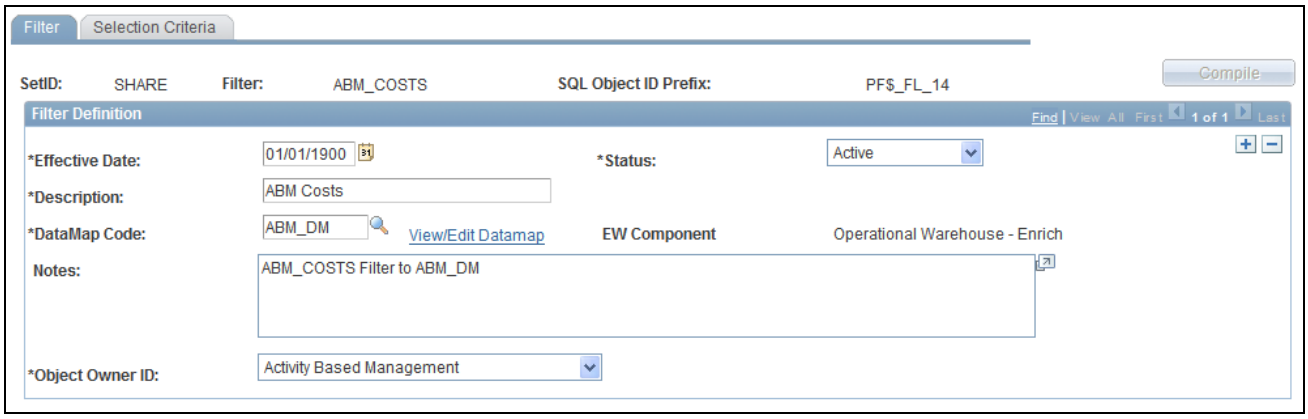
You can create multiple filters based on a single datamap. In addition, your filters, and thus your business rules, can be different for each setID, making the enrichment engines behave differently for the same rules in a different business unit.

Pages Used to Set Up Filters

Page Name	*Definition Name*	*Navigation*	*Usage*
Filter	PF_FILTER_DEFN	EPM Foundation, Business Metadata, Constraint and Expressions, Filter, Filter	Define a filter and specify the datamap.
Filter - Selection Criteria	PF_FILTER_SEQ	EPM Foundation, Business Metadata, Constraint and Expressions, Filter, Selection Criteria	Define selection criteria for business rules.
Tree Viewer	PSTREEVIEWER	Click the Tree button on the Filter - Selection Criteria page. This appears only if you select an operation involving trees.	View all tree nodes. Select a node to be displayed in the value field on the Selection Criteria page by double-clicking the node.

Defining Filters

Access the Filter page (EPM Foundation, Business Metadata, Constraint and Expressions, Filter, Filter).



Filter page

DataMap Code Select the datamap for which you want to define filters.

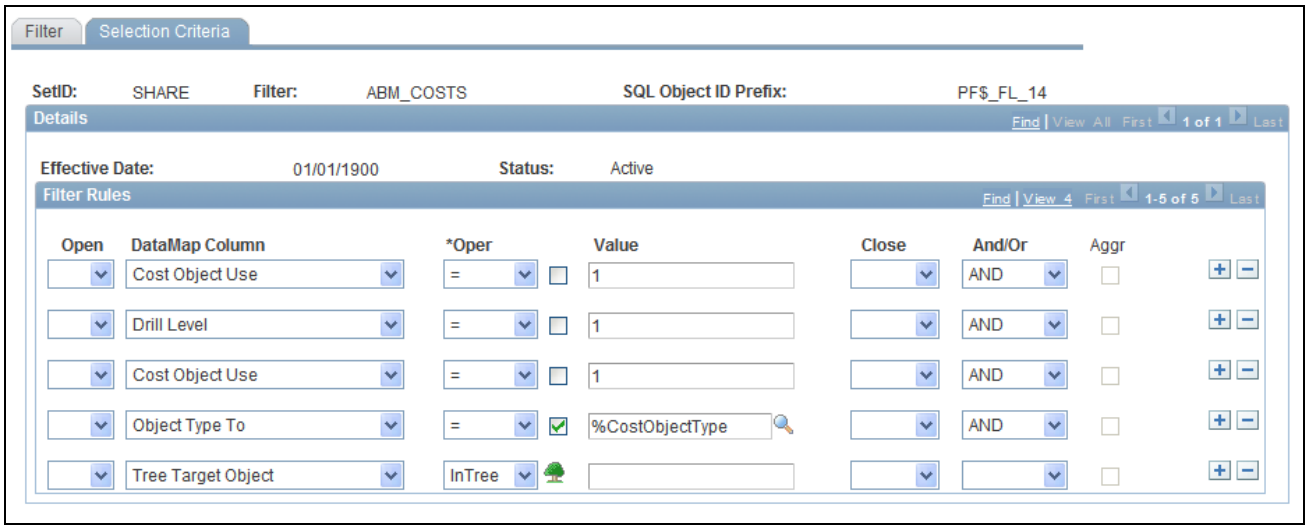
View/Edit Datamap Click to transfer directly to the DataMap page for the selected DataMap to review it or make changes.

Notes Enter any notes to further describe the filter.

When you have finished defining your filter, click the Compile button to compile the filter. This compilation sets the as of date of the filter equal to the effective date. If you want to change the as of date, you must recompile after updating the Effective Date field. Use the Mass Validate utility to validate against as of dates

Specifying Filter Selection Criteria

Access the Filter - Selection Criteria page (EPM Foundation, Business Metadata, Constraint and Expressions, Filter, Selection Criteria).



Selection Criteria page

On this page, you can add or delete actual data values that make up your filter. You can have multiple filter rules.

Open — Select the number of left parentheses needed for the selection criteria. You may have multiple parentheses in a case such as ((X or Y) and Z).

DataMap Column — Select the columns from the datamap for which you want to define filter criteria.

Oper (operation) — Select the operation to be used as selection criteria. The current operations available are *equal to, greater than, less than, greater than or equal to, less than or equal to, not equal to, In Tree, Not in Tree, In MetaTree,* and *NotInMetaTree.*

Like is valid only for character type columns that have no associated lookup tables in the datamap. The value column for this operator must have a pattern search wild card, for example % or _.

If you specify an operation of *In Tree,* click the Tree button that appears alongside the Oper field to specify the tree name and tree node that you want to use as selection criteria. You can search in the Lookup Tree ID page for your tree if it is not displayed. If you still can't find the tree, verify that when you added the tree in the Tree Manager, you also created a record in the Tree Metadata page. This record identifies the engines where the flattened tree data is stored. On selecting the tree name, other information that is related to the tree is retrieved and then sent to the tree viewer (PSTREEVIEWERWRK). All the nodes can be seen here. The node you select here will be displayed in the Value field.

Obj (object) — Select this check box to select from a list of predefined value objects for the value setting. The Obj check box appears if you select any operand other than the tree choices.

Filter Tree — Select this to access the Tree Viewer page and view the tree nodes associated with the tree.

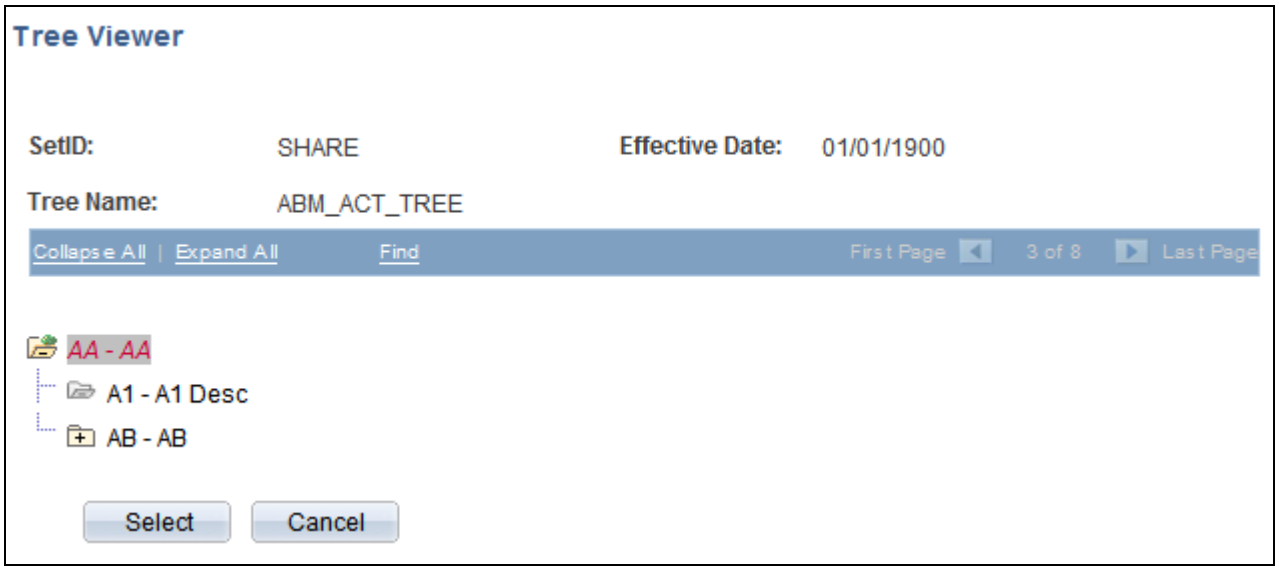
This button only appears if you select one of the tree options for the Operation field.

Value	Select the value that the column is to be evaluated against. If you've defined a lookup table for the column in the datamap, you are prompted to select from a list of valid values.
Close	Select the number of right parentheses that are needed for the selection criteria.
And/Or	Select either an *AND* or an *OR* join for the filter criteria.
Aggr (aggregate)	Select to aggregate the values.

Note. Remember that the Resolver resolves the following fields, so you do not have to include them as selection criteria: setID, business unit, scenario ID, effective date, as of date, fiscal year, and period.

Using the Tree Viewer

Access the Tree Viewer page (Click the Filter Tree button on the Selection Criteria page).



Tree Viewer page

You can expand all levels of the tree to view detailed information. Double-click the folder button to expand specific levels. Select the node of the tree that you want to use and click the Select button to return to the Filter - Selection Criteria page. The value field is populated with the node that you selected.

Setting Up Constraints

Constraints can be made up of one or more filters linked together using AND, OR, and NOT logic. This means that you can keep your filters simple and dedicated to a single purpose, and link them together in constraints to form complex business logic.

Constraints enable you to define business rules for processing and also enable you to create and reuse filters.

This section discusses how to:

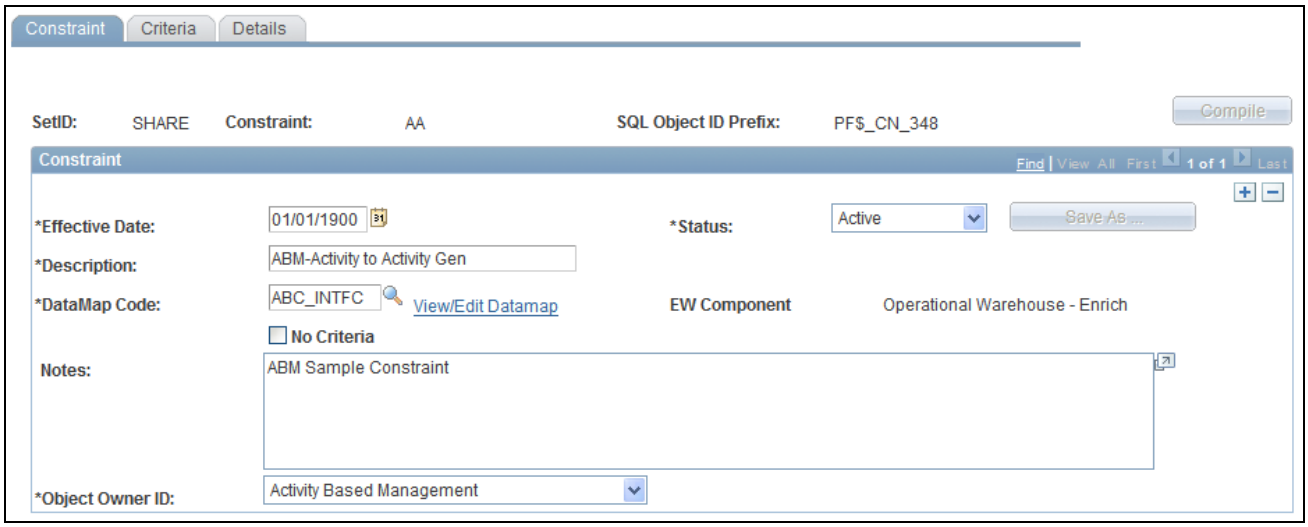
- Define constraints.
- Specify constraint criteria.
- Specify constraint details.

Pages Used to Set Up Constraints

Page Name	*Definition Name*	*Navigation*	*Usage*
Constraint	PF_CONSTRAINT_DEFN	EPM Foundation, Business Metadata, Constraint and Expressions, Constraint, Constraint	Define a constraint.
Constraint - Criteria	PF_CONSTRAINT_SEQ	EPM Foundation, Business Metadata, Constraint and Expressions, Constraint, Criteria	Specify your constraint criteria. This is where you can combine filters to create complex business logic.
Constraint - Details	PF_FILTER_EXPLODE	EPM Foundation, Business Metadata, Constraint and Expressions, Constraint, Details	Display the filter contents of a given constraint. Modify the filter, as needed.

Defining Constraints

Access the Constraint page (EPM Foundation, Business Metadata, Constraint and Expressions, Constraint, Constraint).



Constraint page

Save As Click to replicate existing constraint metadata. You are prompted to enter the new constraint code name and effective date. This is available only in correction mode.

DataMap Code Select the datamap for the constraint.

View/Edit Datamap Click to transfer directly to the DataMap setup page where, you can view and edit the selected datamap.

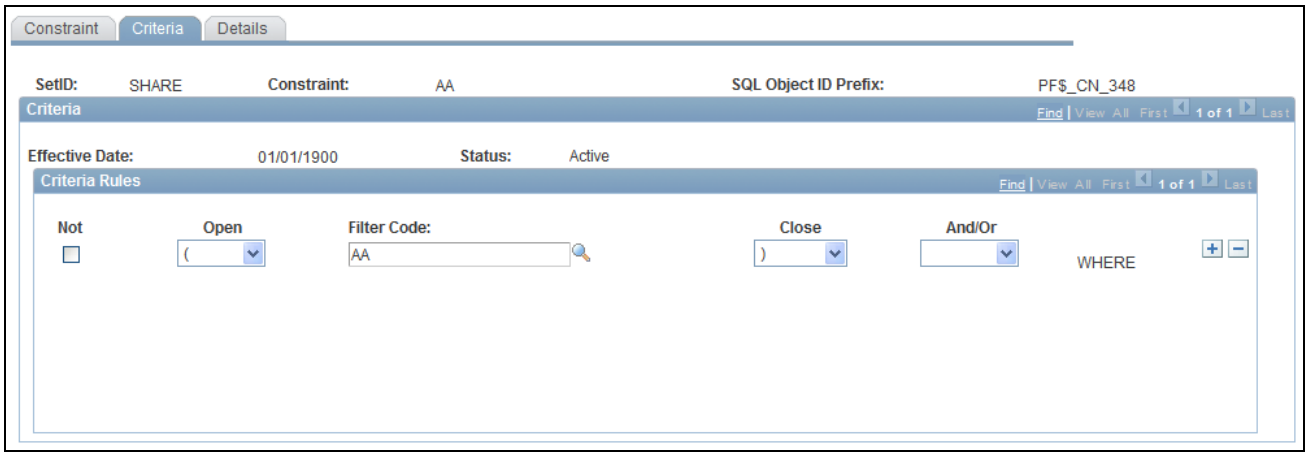
No Criteria You are not required to link filters to your constraint. By selecting this check box, you indicate to the system that you want all the values from the datamap. The system adds a 0= 0 WHERE clause to any other join criteria that the datamap needs (thus hiding the criteria page). The FROM clause is the same as the from field on the datamap.

Notes Enter any notes to further describe the constraint.

When you have completed setup of the constraint, click the Build Constraint SQL button to compile the constraint. This compilation sets the as of date of the constraint equal to the effective date. If you want to change the as of date, you must recompile after updating the Effective Date field. Use the Mass Validate utility to validate against as of dates.

Specifying Constraint Criteria

Access the Constraint - Criteria page (EPM Foundation, Business Metadata, Constraint and Expressions, Constraint, Criteria).



Criteria page

Use this page to add or delete the filters that you want to make up the constraint. You can link one or more filter codes. This page is not accessible if you select the No Criteria check box on the Constraint page.

Not Select to signify the negative of the condition.

Open Select the number of left parentheses that are needed for the selection criteria. You may have multiple parentheses in a case such as ((X or Y) and Z).

Filter Code Select the filter to use. You define filters using the Filter component.

Close Select the number of right parentheses needed.

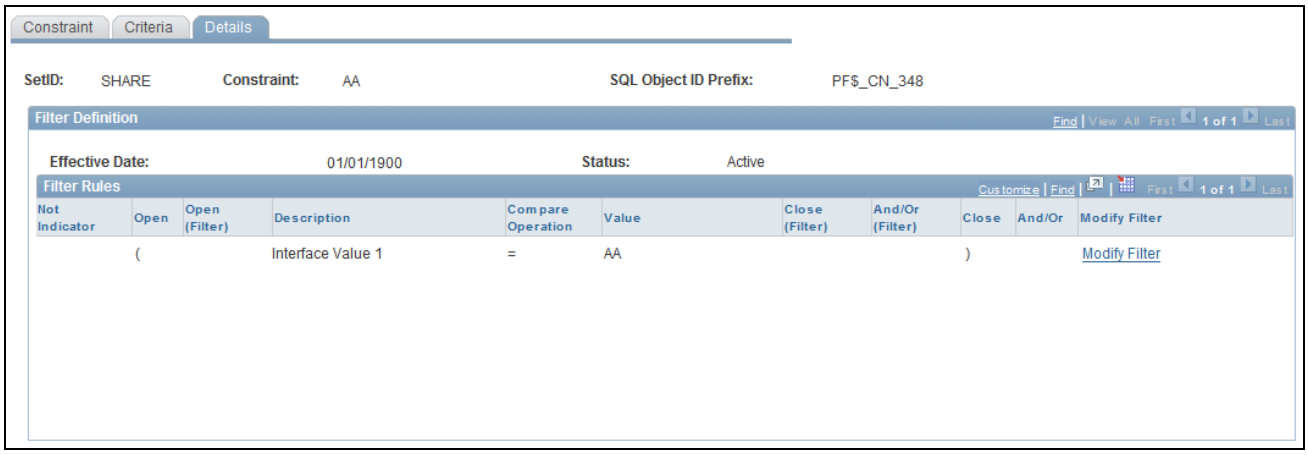
And/Or Select either *AND* or *OR* to relate one line to the next of the filter criteria.

Three filter types are available: *WHERE*, *HAVING*, and a combination of the two. Filter types relate lines of constraint rules.

Note. After a filter is created, the filter type cannot be changed.

Specifying Constraint Details

Access the Constraint - Details page (EPM Foundation, Business Metadata, Constraint and Expressions, Constraint, Details).



Details page

Here you can view the data elements that are targeted based on the filter selection criteria.

Modify Filter Click to access the Filter setup page, where you can modify the filter.

Setting Up Metric Metadata

You can create metric metadata to process specific groups of your transactional data, based on columns in your record tables. These metrics can help you track essential measures for your organization, such as total sales and revenue. PeopleSoft does not deliver metrics. You can, however, create your own metrics using the Define Metric page. The page also enables you to define security for your metric.

See Chapter 6, "Setting Up EPM Security," Defining Dimension and Metric Security, page 128.

Page Used to Set Up Metric Metadata

Page Name	*Definition Name*	*Navigation*	*Usage*
Define Metric	PF_SY_METR_DEFN	EPM Foundation, EPM Security, Metrics and Dimensions, Create/Edit Secured Metrics, Define Metric	Define and secure metrics.

Defining and Securing Your Metrics

Access the Define Metric page (EPM Foundation, EPM Security, Metrics and Dimensions, Create/Edit Secured Metrics, Define Metric).

Define Metric page

Record Name Enter the name of the record that is associated with the metric that you are defining.

Column Name Select a column that is associated with the metric that you are defining.

Is Secured Select this check box to indicate that the metric is secured.

Working with Record Summary Metadata

Record summary metadata specifies the TSE views and pages, as well as the flash total fields that are associated with a table. Record summary metadata is delivered for fact tables. If you change the record structure of a table that uses record summary metadata, or change the totals to be summarized, you must click the Rebuild button to regenerate the SQL. This process is associated with Profit Manager.

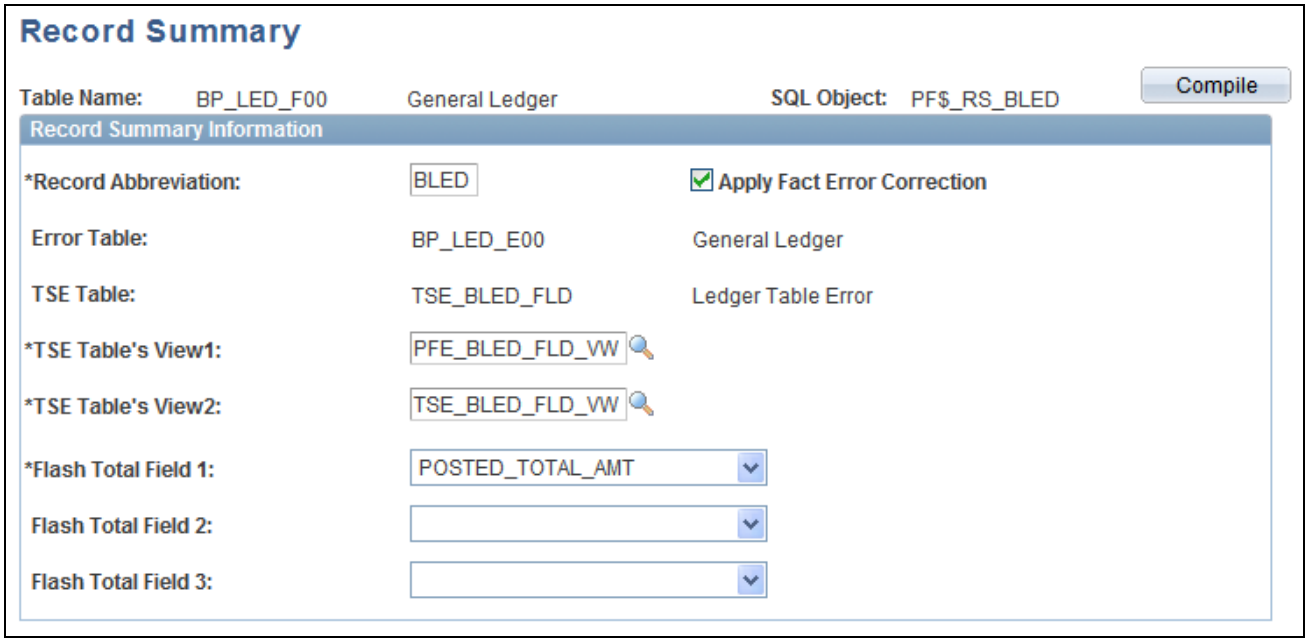
See Chapter 21, "Setting Up and Using Profit Manager," page 497.

Page Used to Work With Record Summary Metadata

Page Name	*Definition Name*	*Navigation*	*Usage*
Record Summary	PF_SUMM_REC_TBL1	EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Record Summary	Create new record summary metadata or rebuild delivered record summary metadata.
PF Record Summary	RUN_PF_SUMM	EPM Foundation, Data Enrichment Tools, Profit Manager, Record Summary, Summarize Error Statistics, PF Record Summary	Run the PF Record Summary engine before you process any fact table modifications.

Creating or Rebuilding Record Summary Metadata

Access the Record Summary page (EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Record Summary).



Record Summary page

SQL Object Prefix ID A system-generated number that identifies the location of the SQL that is built by this component in the SQL Repository.

Record Abbreviation A naming convention that populates all other fields on the page. Enter up to four characters. By defining the record abbreviation, the system makes a best guess and automatically populates all the fields on the rest of the page.

Apply Fact Error Correction Select to apply fact error correction to this table.

Error Table and **TSE Table** Are display-only and based on the error and TSE table definitions that were made in the record metadata for the selected table.

The following fields are automatically populated based on the record abbreviation. These objects are used for PF Modification:

TSE Table's View 1 The work record that is used in the PF Summary process.

TSE Table's View 2 The work record that is used by PF Edi.

Flash Total Field 1, 2, 3 Select the totals that you want to summarize. These totals are defined in the job totals metadata. These are monetary amounts that you want to track.

You can now run the PF Record Summary engine.

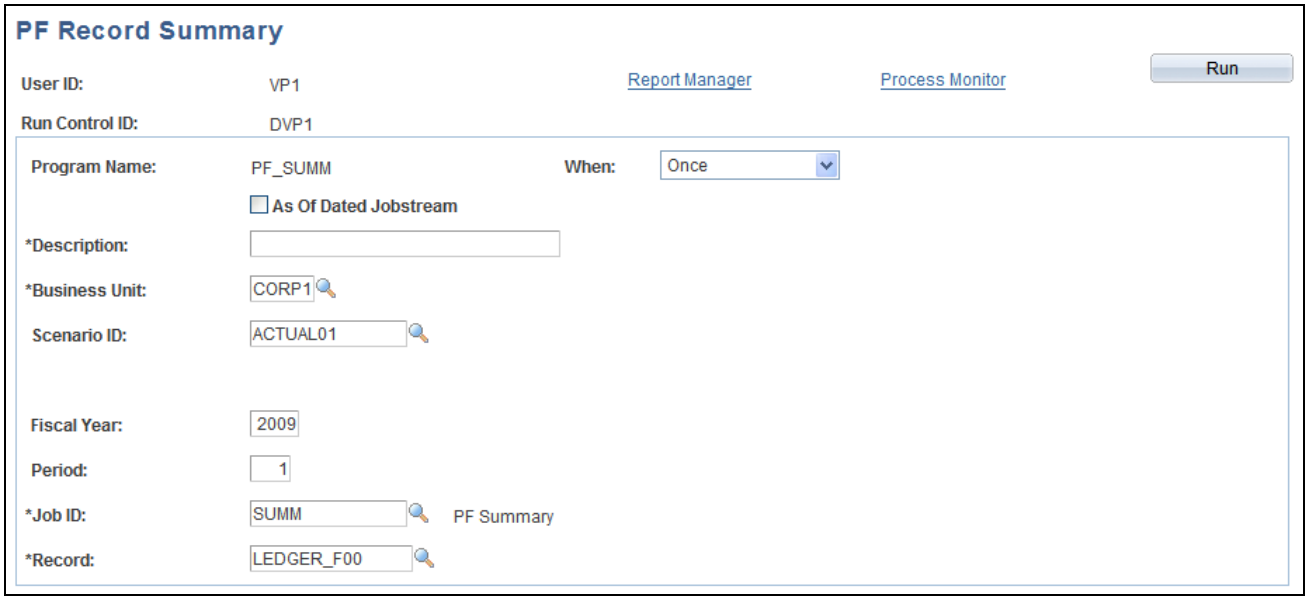
Note. Record Summary metadata must be defined before you run the PF Record Summary engine. You must run the PF Record Summary engine to run PF Modification.

See Also

Chapter 21, "Setting Up and Using Profit Manager," Reviewing and Defining Job Totals Metadata, page 530

Running the PF Record Summary Engine

Access the PF Record Summary page (EPM Foundation, Data Enrichment Tools, Profit Manager, Record Summary, Summarize Error Statistics, PF Record Summary).



PF Record Summary page

Description The description is important because it is used by the Metadata Search engine to locate your metadata.

As Of Dated Jobstream Select to replace the fiscal year and period with the as of date field for the engine run.

Business Unit Select the appropriate business unit.

Scenario ID Select the appropriate scenario.

Fiscal Year Select the appropriate fiscal year. This field does not display for an as of dated jobstream.

Period Select the appropriate period. This field does not display for an as of dated jobstream.

As Of Date If you selected the As of Dated Jobstream check box, enter the date.

Job ID Select the job ID of *Summ.* Job IDs are set up in job metadata.

Record Select the record.

PF_SUMM summarizes error statistics by field and value for the PF Modification process. The record summary metadata uses record metadata to determine which records can be summarized.

Setting Up Report Metadata

Report metadata is delivered for Crystal reports, and delivered PeopleSoft Analytical Application reports. Each delivered report or cube has a record within the Report Metadata page. You can change the metadata as reports change, or as additional parameters are needed. If you create your own reports, you can use report metadata to document them.

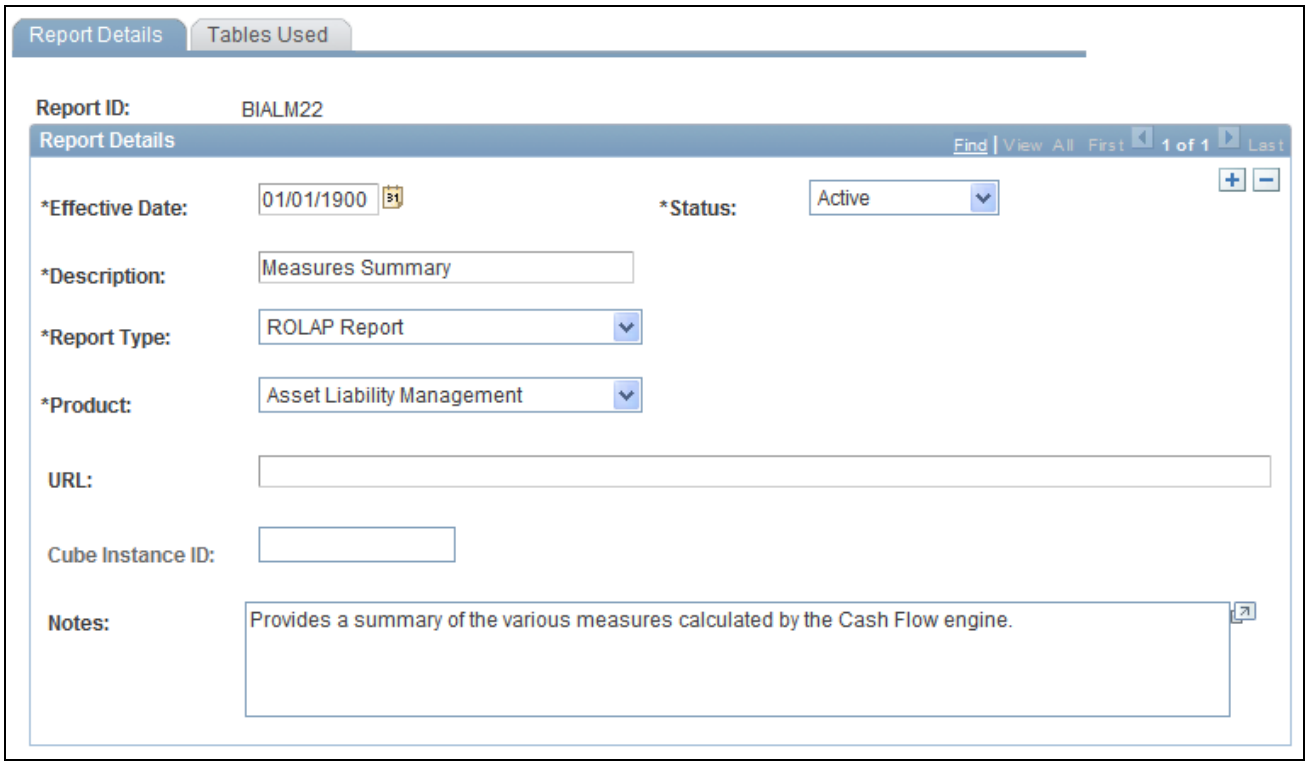
This section discusses how to define report details.

Pages Used to Set Up Report Metadata

Page Name	*Definition Name*	*Navigation*	*Usage*
Report Details	PF_META_RPT_TBL1	EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Report Metadata, Report Details	Define report details, including the report name and type. Report metadata is not required. Rather, it is provided for your reference.
Tables Used	PF_META_RPT_TBL2	EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Report Metadata, Tables Used	Display the tables that are used to generate the report.

Defining Report Details

Access the Report Details page (EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Report Metadata, Report Details).



Report Details page

Report Type The type of report.

Product The associated product for the report.

URL Use to specify either a URL or the directory where the generated reports are stored.

Cube Instance ID The ID assigned by PeopleTools Cube Manager that is used for reporting. This field is active when you select *Cognos Cube* as the report type.

Notes Enter any notes to further describe the report.

Click the Tables Used tab to review the tables that are used to create the selected report.

Cloning Metadata

Cloning metadata is an efficient and quick way to create a duplicate copy of existing or already defined metadata.

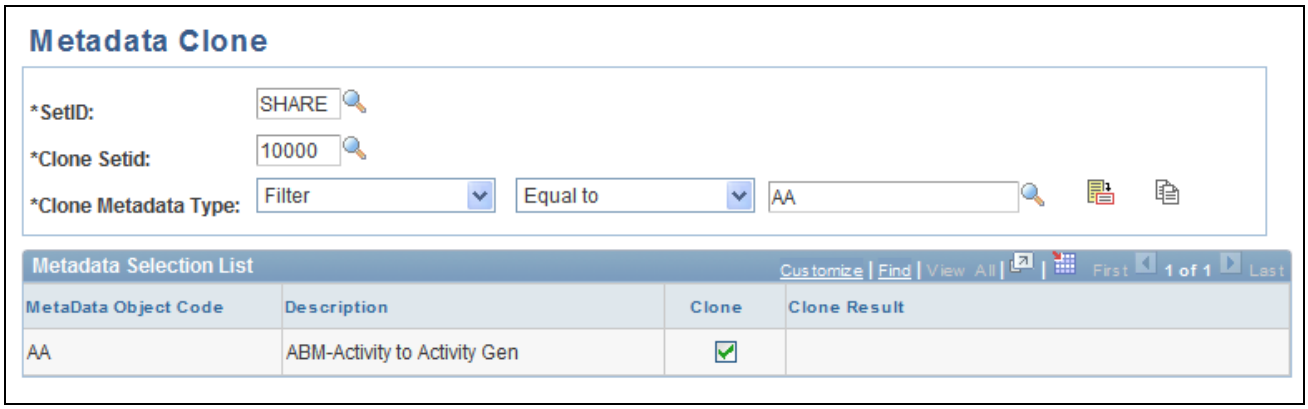
This section discusses how to define metadata clone criteria.

Page Used to Clone Metadata

Page Name	Definition Name	Navigation	Usage
Metadata Clone	PF_MD_CLONE	EPM Foundation, Foundation Metadata, Other Metadata Operations, Metadata Clone	Define metadata clone criteria for copying metadata from one setID to another.

Defining Metadata Clone Criteria

Access the Metadata Clone page (EPM Foundation, Foundation Metadata, Other Metadata Operations, Metadata Clone).



Metadata Clone page

SetID The setID from which you want to clone the metadata.

Clone SetID The setID for which you want to create the clone.

Clone Metadata Type Select the type of metadata that you want to clone. Values are: *AL Rule* (allocation manager rule), *DM Rule* (data manager rule), *Constraint, DataSet*, or *Filter*.

Select on of the following operations: *=, <>, Like,* or *Not Like.*

Enter a metadata value to search on.

Click the Get Selected Metadata button. Your results will appear in the Metadata Selection List field.

Clone Select this check box to return all metadata objects that are based on your selection criteria. Deselect the check box for any metadata that you do not want to clone.

Note. Remember that you must compile all cloned metadata. You can do so using the Mass Compile utility.

See Also

Chapter 17, "Working with Metadata Utilities," Running Mass Compile, page 409

Using the Metadata Mover Utility

This section provides an overview of the Metadata Mover utility and discusses how to:

- Specify metadata export criteria.
- Export metadata from a source to a target database.
- Import metadata into the target database.
- Build imported metadata in the target database.

Understanding the Metadata Mover Utility

The Metadata Mover utility enables you to migrate EPM metadata from one database to another. Specifically, the Metadata Mover utility moves record metadata, tablemaps, datamaps, expressions, filters, constraints, data sets, engine metadata, jobs, jobstreams, report metadata, metric metadata, tree metadata, value objects rule metadata, Data Manager rules, Data Manager rule sets, Allocation Manager rules, Allocation Manager rule sets, Data Mapper rule groups, and Data Mapper rule sets.

The Metadata Mover utility enables metadata to be moved between PeopleSoft 8.8x environments. For example, you cannot move metadata from a PeopleSoft 8.8x environment to a PeopleSoft 8.0x environment. Additionally, to successfully move metadata between PeopleSoft 8.8x EPM environments, the source and target databases must be the same type. For example, if you have a Microsoft database, you must migrate to another Microsoft database. You can't migrate from a Microsoft database to an Oracle database.

The Metadata Mover utility maintains SQL counters for the target database. The source SQL counter will be lost when the move is completed so that uniqueness in ensured within the target database.

Follow these steps to move EPM metadata from your source database to your target database:

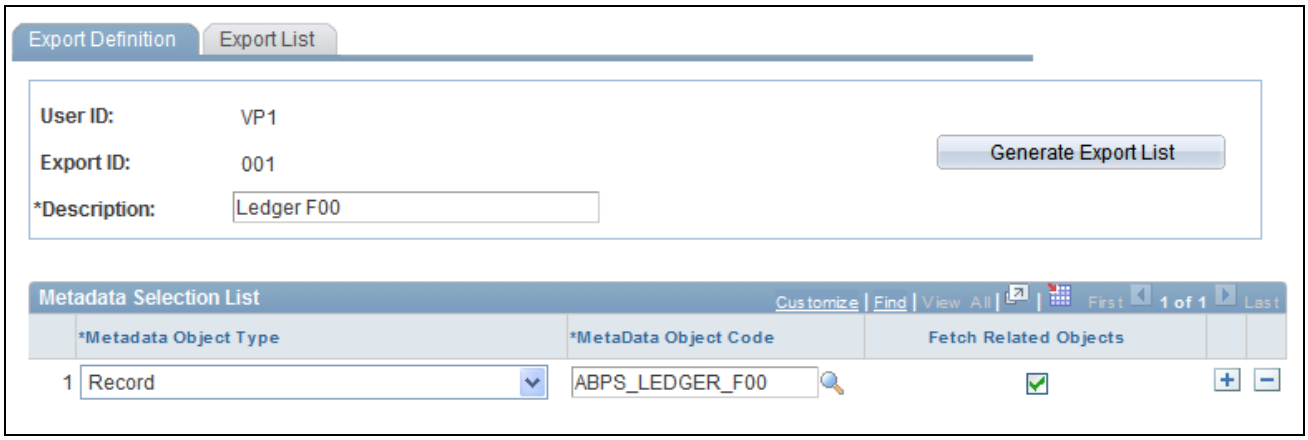
1. Sign in to your source database.
2. Select the metadata that you plan to export from your source database using the Metadata Export page.
3. Use the Data Mover script to export the objects from the source database.
4. Use the Data Mover script to import the objects into the target database.
5. Use the Mass Compile and Audit processes in the target database to build and validate the migrated metadata objects.

Pages Used to Migrate Metadata

Page Name	*Definition Name*	*Navigation*	*Usage*
Metadata Export - Export Definition	PF_MD_EXPORT	EPM Foundation, Foundation Metadata, Other Metadata Operations, Metadata Export, Export Definition	Specify metadata export criteria.
Metadata Export - Export List	PF_MD_EXPLST	EPM Foundation, Foundation Metadata, Other Metadata Operations, Metadata Export, Export List Click the Generate Export List button on the Export Definition page.	View a list of metadata objects that you selected on the Export Definition page. These objects will be exported to your target database. The list includes the following information for each metadata object: metadata type, setID value, metadata object code, and related objects.

Specifying Metadata Export Criteria

Access the Export Definition page (EPM Foundation, Foundation Metadata, Other Metadata Operations, Metadata Export, Export Definition).



Export Definition page

Note. You can maintain many export IDs, although you can have only one metadata export list. This means that when you generate the metadata export list, the system replaces the existing list with a new list.

In the Metadata Selection List group box, select:

Metadata Object Type and **Metadata Object Code**
Select each metadata object type and code that you want to export to your target database. Remember that you must select metadata dependencies (if the metadata has dependencies).

Fetch Related Objects
Select this check box to have the system fetch all objects that are related to or dependent on the object that you have selected and add them to the metadata export list.

Generate Export List
Click to have the system generate the list of objects to be exported. You can view them on the Metadata Export - Export List page.

For example, if you select a datamap as the object type and select the Fetch Related Objects check box, the system adds all related expressions, filters, constraints, data sets, Data Manger rules, Allocation Manager rules, and value objects to the metadata export list.

The following table shows which metadata objects are dependent upon each other.

Metadata Type	***Related Objects***
Allocation Manager rules	Meta Values.
Allocation Manager rule set	Allocation Manager Rules and Meta Values.
Data Manager rules	Meta Values.
Data set	Option is not available.
Data Manager rule set	Data Manager rules and meta values.
Data Mapper rule set	Set of data mapper rules.
Data Mapper rule group	Group of Data Mapper rules.
Constraint	Data sets, Allocation Manager rules, Allocation Manager meta values, Data Manager rules, and Data Mart meta values.
Filter	Constraints and related objects.
Datamap	Expression, filter, and related objects.
Tablemap	Datamap and related objects.

Metadata Type	*Related Objects*
Record	Tablemap and related objects.
Jobstream	Option is not available.
Job	Jobstream.
Engine	Job and jobstream.
Tree, metric, report, and rule metadata, expressions, Data Loader and data mart data	Option is not available.

Review the metadata export list and if you are sure that this is the data that you want to export, proceed to the next step.

Note. Data Mapper is documented in the *PeopleSoft Enterprise Global Consolidations PeopleBook.*

See *PeopleSoft Enterprise Global Consolidations 9.1 PeopleBook*, "Preparing Data for Consolidations," Mapping to a Common Chart of Accounts (Data Mapper).

Exporting Metadata from a Source Database to a Target Database

After selecting the metadata that you want to export from your source database to your target database (using the Export Definition page), perform the export. To perform the export, follow these steps:

1. Sign in to Data Mover in the metadata source database.
2. Open PFMMEXP.dms in the Data Mover tool.
3. Edit the path name for the output files (you can select which path you want).
4. Run the script.

   The script exports the metadata tables into a DAT file for the export list that is generated through the Export Definition page.
5. Sign out of the source database.

You now need to import the metadata that you just exported.

Importing Metadata into the Target Database

Now that you have exported the metadata from the source database, you must import the metadata into the target database in the following way:

1. Sign in to the target database.

2. Open PFMMIMP.dms in the Data Mover tool.
3. Edit the path of both the input files to point to the output directory of the exported files.
4. Run the script.

   The script exports the metadata tables from the .DAT files.

Building Imported Metadata in the Target Database

After you've finished importing the metadata into your target database, you must build the new metadata. To build the new imported metadata, you must first recompile it. If you imported a small number of new metadata objects, you can go to each metadata object's page and click the Compile button.

Note. Now there is no SQL object counter on the pages. Clicking the Compile button generates the counter.

If you have imported a large quantity of metadata objects, run the Mass Compile process followed by the Audit process.

Correct any errors that were identified during the Audit process.

Rerun the audit until it is clean.

See Also

Chapter 17, "Working with Metadata Utilities," Auditing PeopleSoft EPM Objects, page 403

Chapter 17, "Working with Metadata Utilities," Running Mass Compile, page 409

Deleting Metadata

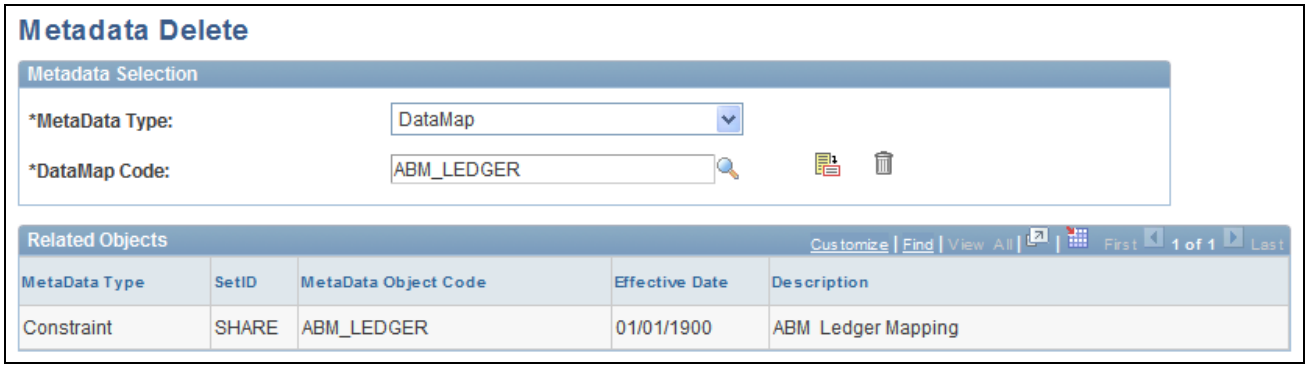
The Metadata Delete process enables you to delete records, tablemaps, datamaps, and rule metadata.

Page Used to Delete Metadata

Page Name	*Definition Name*	*Navigation*	*Usage*
Metadata Delete	PF_METADATA_DEL	EPM Foundation, Foundation Metadata, Other Metadata Operations, Delete Metadata, Metadata Delete	Delete metadata.

Deleting Metadata

Access the Metadata Delete page (EPM Foundation, Foundation Metadata, Other Metadata Operations, Delete Metadata, Metadata Delete).



Metadata Delete page

MetaData Type Select the type of metadata that you want to delete.

Metadata Object Code Based on the metadata type that you have selected, select the metadata object from the valid values that are available. For example, if you select datamap as the metadata type, you will be able to select from the available datamap objects.

Click the Fetch Related Objects button to populate the grid with the related metadata objects.

Click the Delete All Objects button to delete the object code and all the related objects.

Creating User-Defined Functions

User-defined functions enable you to define functions one time through a common interface, then use them throughout many of the Analytical Applications. The options that are available to you when defining functions are based on predefined modules that are provided with your system.

Refer to the *PeopleSoft Application Fundamentals for Financial Services Industry Applications* for details on working with user-defined functions.

See Also

PeopleSoft Enterprise Application Fundamentals for Financial Services Industry 9.1 PeopleBook, "Creating User-Defined Functions"

OWE Dimension Correction Pages

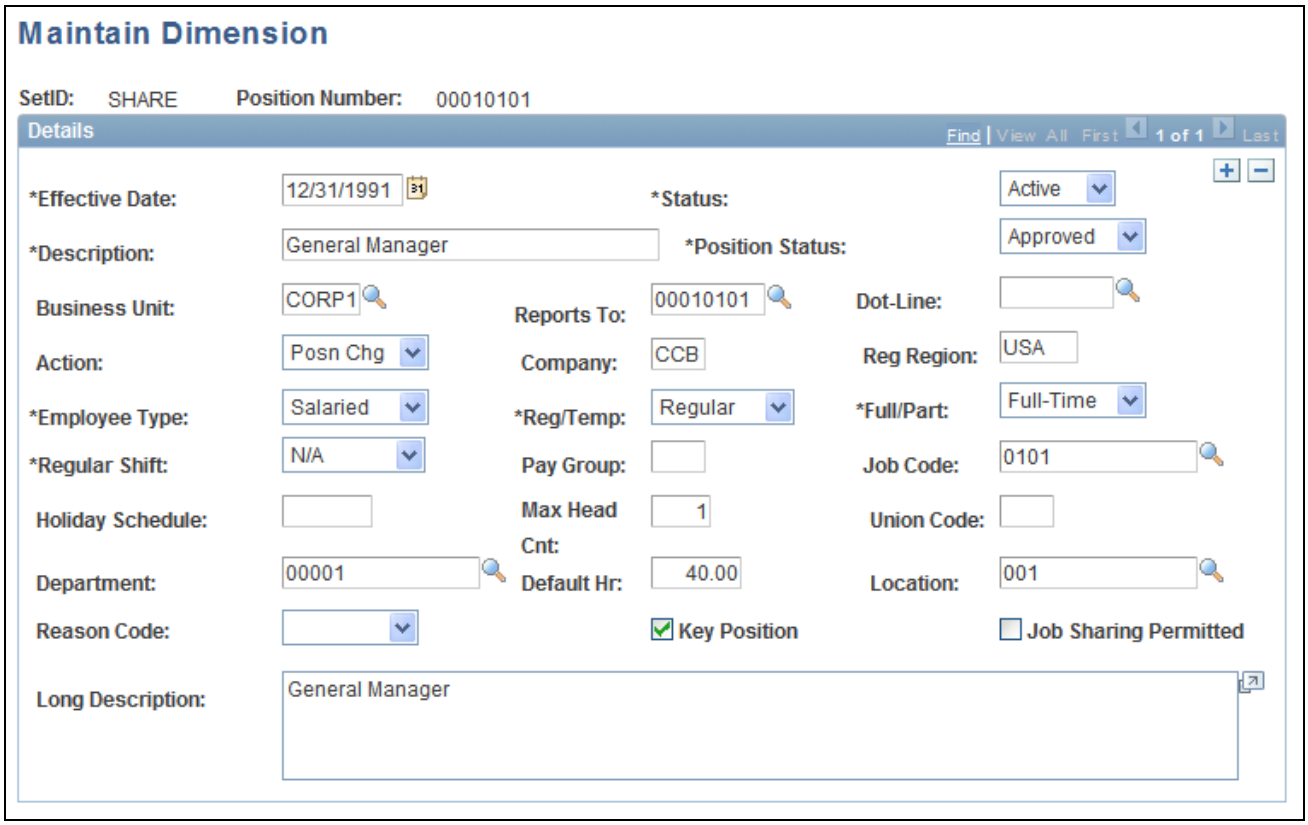
Organizational change is both necessary and inevitable. Your organization will restructure departments, hire new department managers, develop new products, and obtain new customers through a variety of sales channels. In each case, you must capture these business rules in EPM.

Although the prepackaged ETL jobs are designed to automatically capture these changes to your organization, you might encounter times when you are required to add or update dimensions directly in the PeopleSoft pages. You can use the OWE Dimension Maintenance menu items for this purpose (EPM Foundation, Business Metadata, OWE Dimension Maintenance).

This menu includes a separate menu item for each functional area that corresponds to one or more dimensions from your PeopleSoft source data and contains the following categories:

- Common.
- CRM.
- Supply Chain Management.
- Human Resource Management System.

For example, under the Common menu there is a specific menu item to add or update common dimensions. The following is an example of one of the maintain dimension pages:



Maintain Dimension page

For each dimension, enter the relevant information for your dimension and save.

Warning! It is recommended that dimensions be accessible only to the EPM Administrator.

Note. Critical errors, such as missing setIDs or business units, must be fixed on their respective setup pages. Typically, the types of errors you correct in the dimension pages are the non-key fields of an error record.

Chapter 17

Working with Metadata Utilities

This chapter discusses how to:

- View error messages.
- Audit PeopleSoft EPM objects.
- Perform impact analysis.
- Run Mass Validate.
- Run Mass Compile.
- Work with tree utilities.
- Set up and flatten tree metadata.

Viewing Error Messages

You can view error messages generated by the engines described in this chapter using the Engine Messages component. This component is described in the "Streamlining Processing with Jobstreams" chapter of this PeopleBook.

See Chapter 20, "Streamlining Processing with Jobstreams," Viewing Engine Messages, page 494.

Auditing PeopleSoft EPM Objects

This section provides an overview of the Audit utility and discusses how to:

- Run the Audit utility.
- Review metadata audit log error messages.

Understanding the Audit Utility

The PeopleSoft Enterprise Performance Management (PeopleSoft EPM) Audit utility is a common process used to identify setup errors in PeopleSoft EPM modules. For instance, you can use the Audit utility to check the validity and existence of metadata objects in PeopleSoft EPM application tables. The Audit utility verifies that your original metadata objects (tablemaps, datamaps, constraints, filters, trees, and so forth) are still intact. The Audit utility also provides audit functionality for row-level security group objects. To remove constraints, value objects, SQL object IDs, and counter objects from the row level group tables, you must fix the security group or the missing object manually.

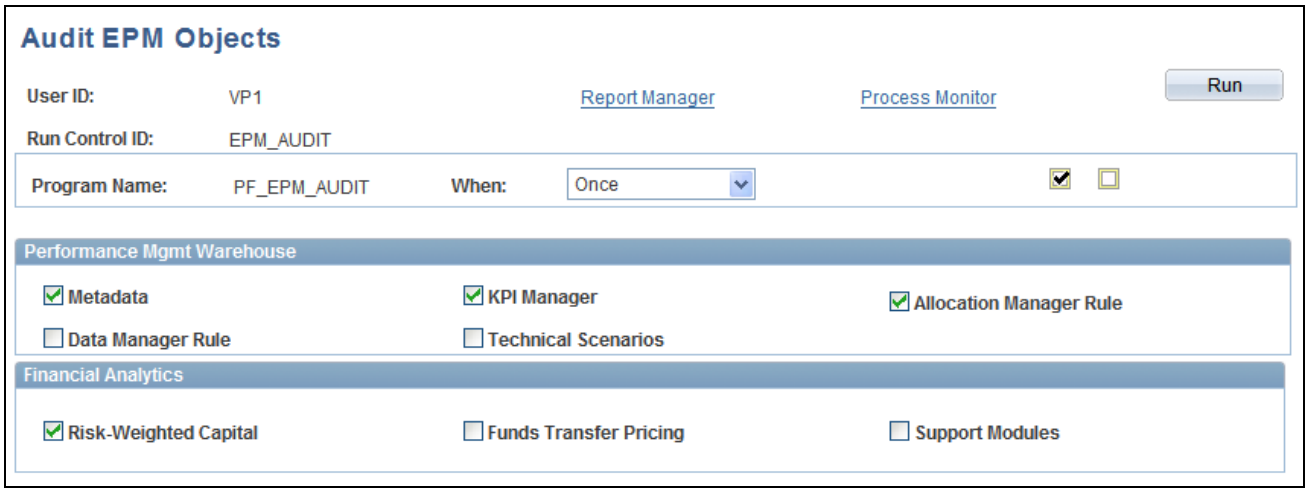
The Audit utility process does not need to be set up in a jobstream and can be run at any time. The processing time depends on how many selections you make. This utility can affect database performance times. To ensure a successful engine run, PeopleSoft suggests that you run an audit, review results using the Process Monitor and Messages inquiry page, and fix any problems before running a jobstream. Any problems identified by the Audit utility may require assistance from a programmer or warehouse administrator.

Page Used to Audit PeopleSoft EPM Objects

Page Name	*Definition Name*	*Navigation*	*Usage*
Audit EPM Objects	RUN_PF_AUDIT	EPM Foundation, Foundation Metadata, Other Metadata Operations, Audit EPM Objects	Run the Audit utility to identify setup errors.

Running the Audit Utility

Access the Audit EPM Objects page (EPM Foundation, Foundation Metadata, Other Metadata Operations, Audit EPM Objects).



Audit EPM Objects page

Click the Select All button to run all of the engine options available.

Click the Clear All button to deselect all check boxes.

Performance Mgmt Warehouse (Performance Management Warehouse)

In this group box the audit options are: Metadata, KPI Manager, Allocation Manager Rule, Row Level Security, Data Manager Rule, and Technical Scenarios.

The metadata objects that are examined by the Audit utility include record metadata, tablemaps, datamaps, constraints, filters, expressions, data sets, engines, jobs, and jobstream rules.

If you select the Data Manager Rule check box, the Audit utility checks for inconsistencies between data manager rules and the metadata.

KPI Manager objects that are examined by the Audit include datamaps, data elements, dimensions, constraints, filters, calculation expressions, calculation cubes, and calculation IDs.

Financial Analytics

Here you can select to run the audit on Risk-Weighted Capital, Funds Transfer Pricing, or Support Modules.

Running the Audit Utitlity

Click the Run button to run this request.

After the audit has run, you can check the Process Monitor to verify the process ran smoothly. Use the Messages inquiry pages to review any errors found by the audit.

Note. You can run the Audit utility at any time. The processing time depends on how many selections you make and depending on your selections, the utility can affect database performance times.

Reviewing Metadata Audit Log Error Messages

All messages can be found and viewed by navigating to PeopleTools, Utilities, Administration, Message Catalog. EPM messages are numbered 9000-9999.

Performing Impact Analysis

Impact Analysis enables you to determine the effects that a change to one PeopleSoft EPM object may have on the other objects related to it. For instance, depending on the type of change you make to a warehouse table, you may have to change the tablemaps and datamaps built on top of the table. Impact Analysis provides a list of the objects that are related to the object that you want to change so that you can determine the impact of the change.

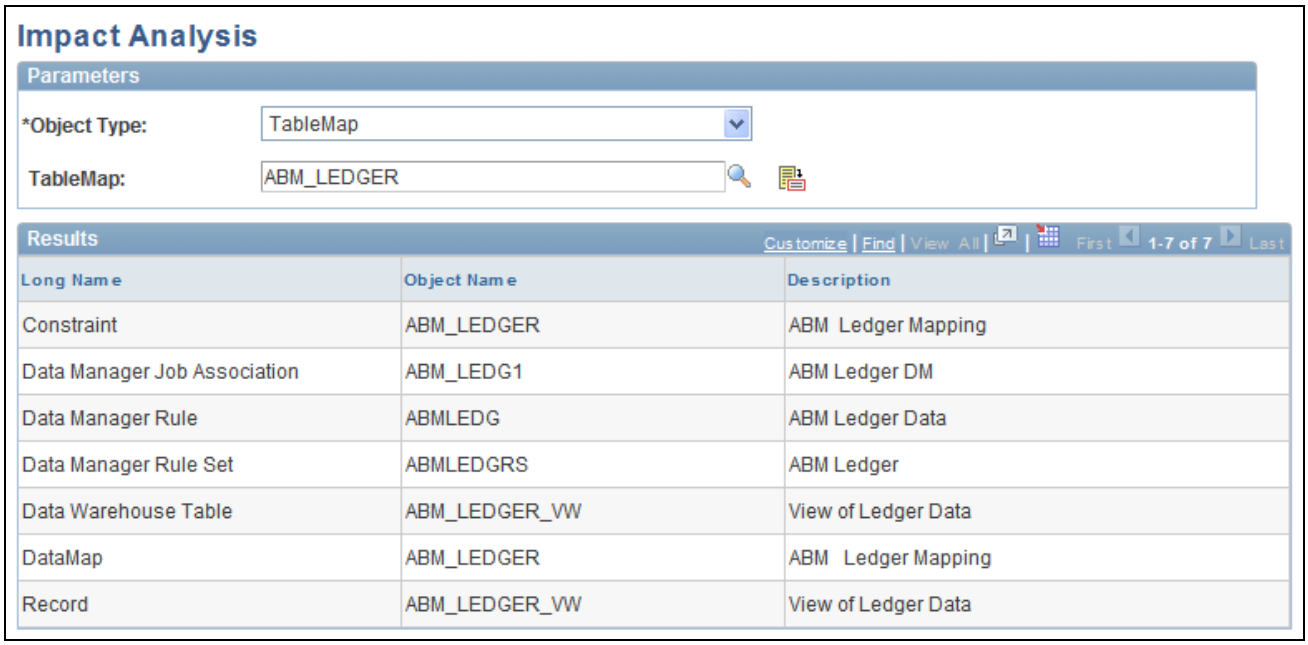
This section discusses how to use Impact Analysis on PeopleSoft EPM objects.

Page Used to Perform Impact Analysis

Page Name	Definition Name	Navigation	Usage
Impact Analysis	PF_IMPACT_REQUES2	EPM Foundation, Foundation Metadata, Other Metadata Operations, Impact Analysis	Determine the relationships between data warehouse objects.

Using Impact Analysis

Access the Impact Analysis page (EPM Foundation, Foundation Metadata, Other Metadata Operations, Impact Analysis).



Impact Analysis page

Object Type Select the type of object you want to review. The field below changes based on the object type you select. For example, if you select *DataMap*, the field label changes to DataMap.

Enter the name of the object.

Click the Related Objects button to populate the grid with all the objects related to the object you specified.

You can download the list of objects using the download button at the top of the grid. Using this list, you decide whether a change is feasible. You can also use the list to update team members when something changes.

Note. This is a display-only page that provides a list of all related objects.

Running Mass Validate

The Mass Validate metadata utility enables you to validate, but not compile, metadata objects. Mass Validate will certify all as of dates created for filters, constraints, and data sets for the specified run date. This utility helps ensure your metadata is valid at run time and increases your chance of a successful engine run.

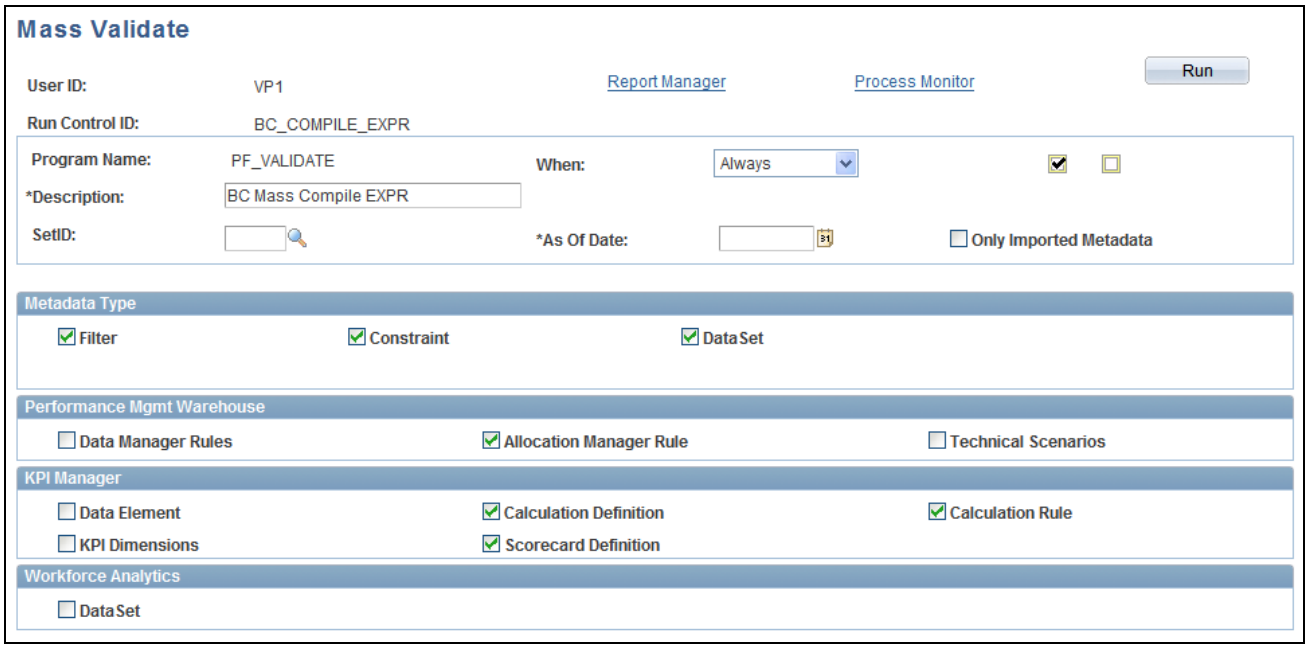
This section discusses how to validate metadata objects.

Page Used to Run Mass Validate

Page Name	*Definition Name*	*Navigation*	*Usage*
Mass Validate	RUN_PF_VALIDATE	EPM Foundation, Foundation Metadata, Other Metadata Operations, Mass Validate	Validate metadata objects.

Validating Metadata Objects

Access the Mass Validate page (EPM Foundation, Foundation Metadata, Other Metadata Operations, Mass Validate).



Mass Validate page

	Click the Select All button to run all of the engine options available.
	Click the Clear All button to deselect all check boxes.
SetID	Select the setID you want to run validate on.
As Of Date	Specify as of date for this run.
Only Imported Metadata	Select this check box to limit Mass Validate to only those objects which have their SQL counter field set to zero. Use this to validate any metadata imported into the database using the Metadata Migration utility. This metadata will not have a SQL ID when first imported. Also use it to validate any new metadata that has not been compiled. First validate the metadata and then run Mass Compile to compile it.

You can either select all objects, or individually select the objects you want to validate in the Metadata Type, Performance Management Warehouse, KPI Manager, and Workforce Analytics group boxes.

Click Run to run the Mass Validate process.

You can review any errors generated by the Mass Validate process by creating an engine error log.

See Also

Chapter 21, "Setting Up and Using Profit Manager," Creating Error Log Reports for the Performance Ledger, page 506

Running Mass Compile

The Mass Compile metadata utility enables you to either compile individual metadata objects, or all metadata objects.

As discussed in the "Setting Up and Working with Metadata" chapter of this PeopleBook, if you change a table, you must recompile record metadata for that table. For instance, if you add a non-key column to a table, you must recompile the record metadata. If you add a key column, you must recompile both the record metadata and any tablemaps, datamaps, constraints, or other metadata objects associated with it. The advantage of Mass Compile is that you can opt to compile all metadata objects at once. In addition, if you have imported metadata into the database and validated it, you can compile it using Mass Compile.

Note. Mass Compile will not compile allocation manager rules that are period-based or have the multiple business unit option selected. These rules will be skipped and must be compiled from the Allocation Manager Rules component.

This section discusses how to compile metadata objects using Mass Compile.

See Also

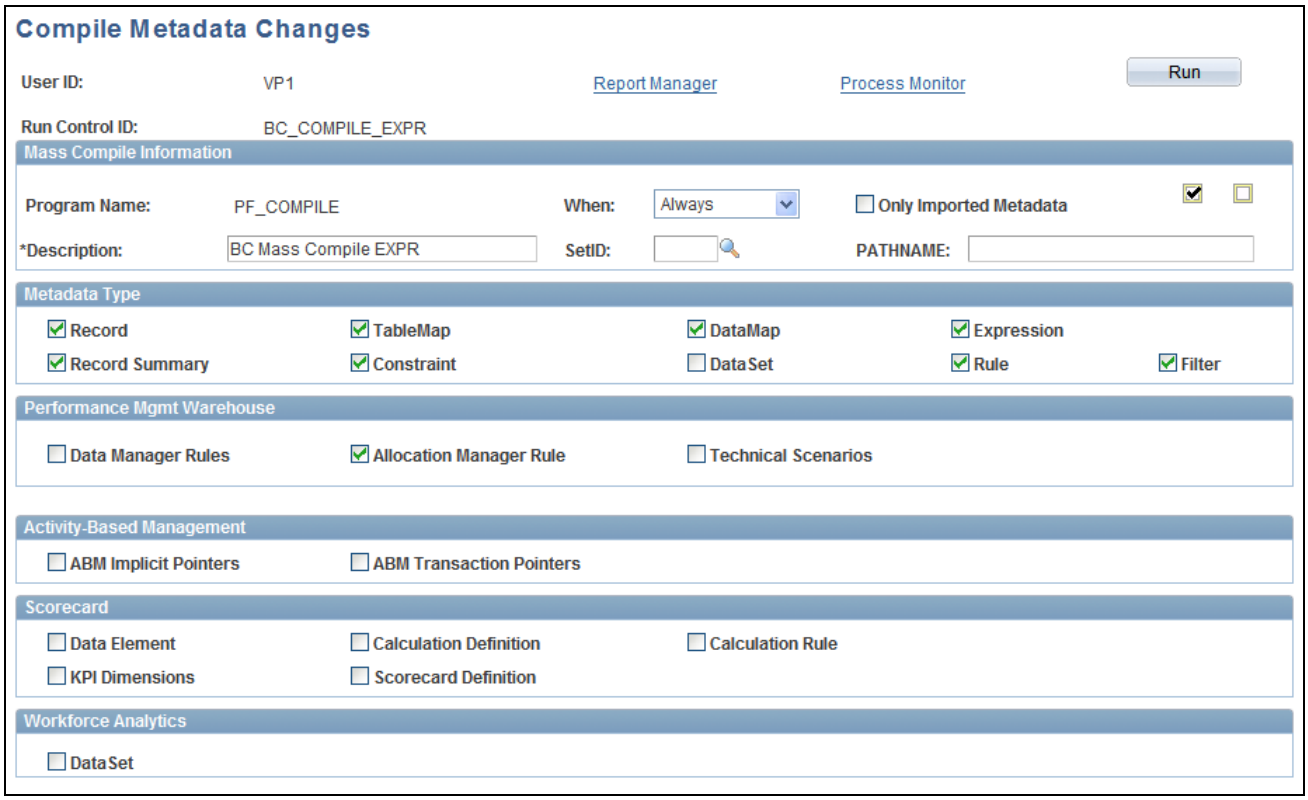
Chapter 22, "Using Data Enrichment Tools," Defining Allocation Manager Rules, page 572

Page Used to Run Mass Compile

Page Name	*Definition Name*	*Navigation*	*Usage*
Compile Metadata Changes	RUN_PF_COMPILE	EPM Foundation, Foundation Metadata, Other Metadata Operations, Compile Metadata Changes	Compile metadata objects by running Mass Compile. To view errors, see select the Process Monitor or Report Manager link.

Compiling Metadata Objects using Mass Compile

Access the Compile Metadata Changes page (EPM Foundation, Foundation Metadata, Other Metadata Operations, Compile Metadata Changes).



Compile Metadata Changes page

	Click the Select All button to select all the metadata types listed.
	Click the Clear All button to deselect all check boxes.
SetID	Select the setID to limit the compile to the specified input setID.
Only Imported Metadata	Select this check box to limit the Mass Compile utility to only those objects with zero SQL counter that have been imported into the database and need compiling. You should validate this metadata beforehand using the Mass Validate process.
PATHNAME	Enter the directory location where you would like the output log to be placed.

You can either select all objects, or individually select the objects you want to validate in the Metadata Type, Performance Management Warehouse, Activity-Based Management, Scorecard, and Workforce Analytics group boxes.

Working With Tree Utilities

This section provides an overview of tree utilities and discusses:

- Set up Tree Compare and run the Tree Compare job.
- Review Tree Compare results.

- Set up and run Super Tree.
- Review Super Tree results.

Understanding Tree Utilities

There are two types of tree utilities provided with EPM , Tree Compare and Super Tree.

Tree Compare Utility

The Tree Compare utility enables you to compare effective dates for trees. The results page shows nodes that have been added, deleted, or moved from one parent to another. You may also view the detail objects that have changed. To perform a tree compare, complete the Tree Compare setup page, run the Tree Compare job in a jobstream, and then inquire on the Tree Compare output using the Tree Compare inquiry page.

Super Tree Utility

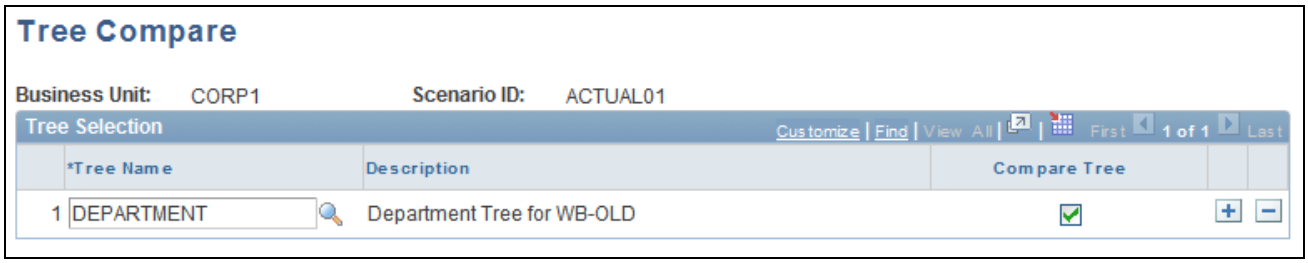
The Super Tree utility enables you to combine multiple effective dates of a tree into one. This super tree contains all tree changes for a certain period of time enabling you to analyze effective-dated trees more easily.

Pages Used to Work With Tree Utilities

Page Name	*Definition Name*	*Navigation*	*Usage*
Tree Compare	PF_TREE_CMP_TBL1	EPM Foundation, Business Metadata, Tree Metadata, Setup Tree Compare, Tree Compare	Set up and run Tree Compare by selecting trees to be compared for a business unit and scenario.
Tree Compare inquire page	PF_TREECOMP1	EPM Foundation, Business Metadata, Tree Metadata, Setup Tree Compare, Review Tree Compare, Tree Compare	Inquire on the results of a tree compare.
Super Tree	RUN_PF_SUP_TREE	EPM Foundation, Business Metadata, Tree Metadata, Update Super Tree, Super Tree	Merge multiple trees into one tree.
Super Tree inquire page	PF_SUP_TREE1	EPM Foundation, Business Metadata, Tree Metadata, Review Super Tree, Super Tree	Review results of the Super Tree process.

Setting up Tree Compare and Running the Tree Compare Job

Access the Tree Compare page (EPM Foundation, Business Metadata, Tree Metadata, Setup Tree Compare, Tree Compare).



Tree Compare page

Tree Name Select the tree you want to compare. A description displays.

Compare Tree Select this check box to compare the tree. If this check box is not selected, Tree Compare does not include this tree.

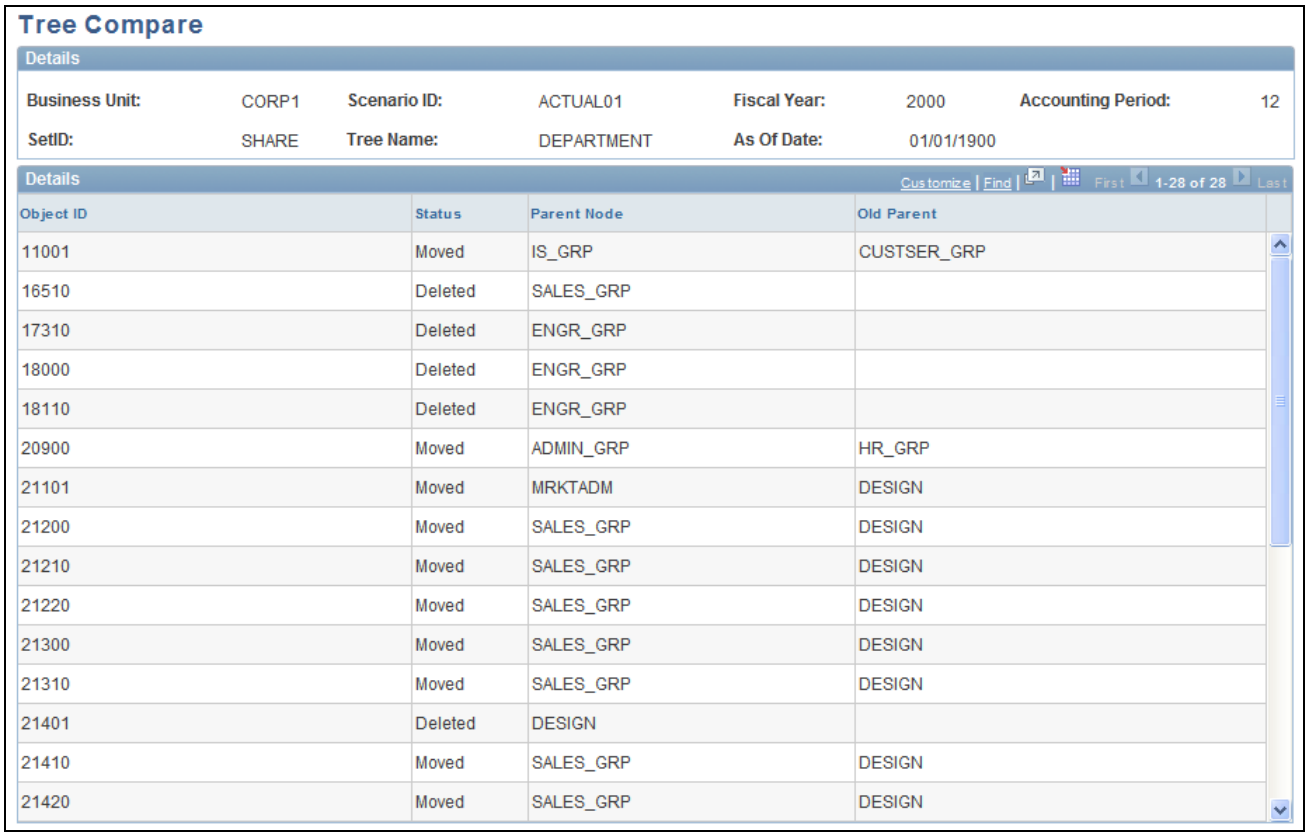
You can now run the Tree Compare.

To run the Tree Compare job, include the job PF_TREECOMP job in a jobstream.

Once the job has run, review the results on the Tree Compare inquire page.

Reviewing Tree Compare Results

Access the Tree Compare inquire page (EPM Foundation, Business Metadata, Tree Metadata, Setup Tree Compare, Review Tree Compare, Tree Compare).



Tree Compare

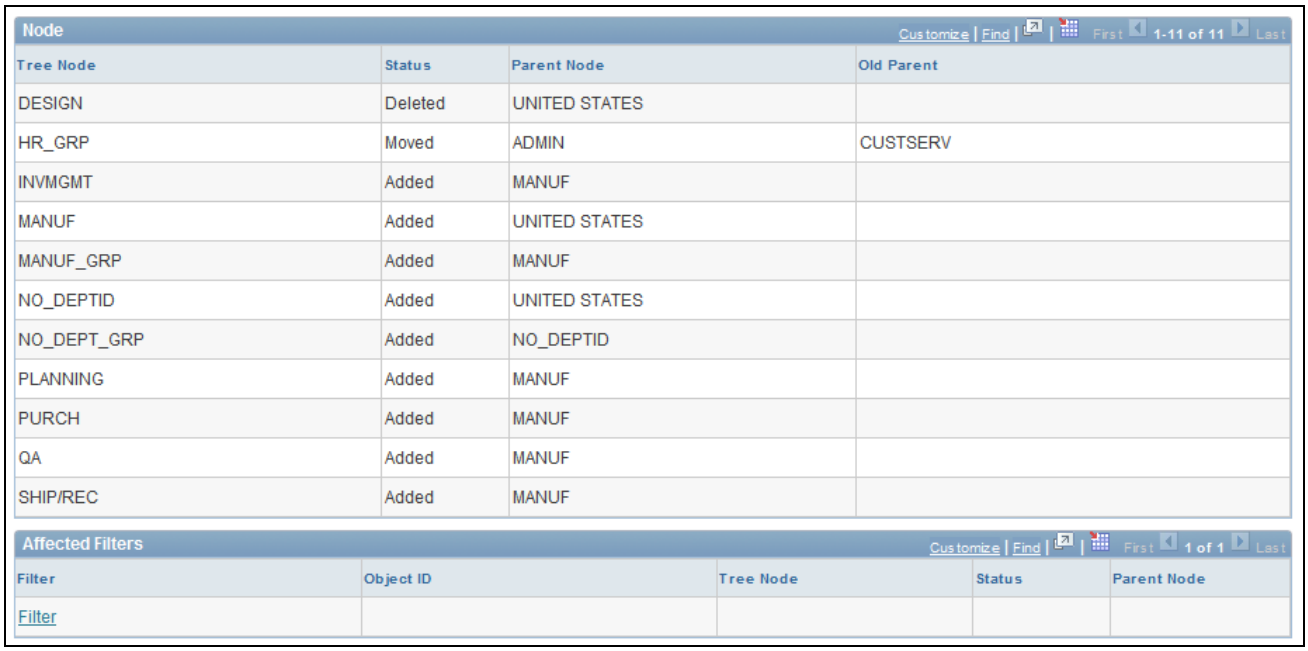
Details

Business Unit:	CORP1	Scenario ID:	ACTUAL01	Fiscal Year:	2000	Accounting Period:	12
SetID:	SHARE	Tree Name:	DEPARTMENT	As Of Date:	01/01/1900		

Details — Customize | Find | First 1-28 of 28 Last

Object ID	Status	Parent Node	Old Parent
11001	Moved	IS_GRP	CUSTSER_GRP
16510	Deleted	SALES_GRP	
17310	Deleted	ENGR_GRP	
18000	Deleted	ENGR_GRP	
18110	Deleted	ENGR_GRP	
20900	Moved	ADMIN_GRP	HR_GRP
21101	Moved	MRKTADM	DESIGN
21200	Moved	SALES_GRP	DESIGN
21210	Moved	SALES_GRP	DESIGN
21220	Moved	SALES_GRP	DESIGN
21300	Moved	SALES_GRP	DESIGN
21310	Moved	SALES_GRP	DESIGN
21401	Deleted	DESIGN	
21410	Moved	SALES_GRP	DESIGN
21420	Moved	SALES_GRP	DESIGN

Tree Compare page, 1 of 2



Node

Customize | Find | First 1-11 of 11 Last

Tree Node	Status	Parent Node	Old Parent
DESIGN	Deleted	UNITED STATES	
HR_GRP	Moved	ADMIN	CUSTSERV
INVMGMT	Added	MANUF	
MANUF	Added	UNITED STATES	
MANUF_GRP	Added	MANUF	
NO_DEPTID	Added	UNITED STATES	
NO_DEPT_GRP	Added	NO_DEPTID	
PLANNING	Added	MANUF	
PURCH	Added	MANUF	
QA	Added	MANUF	
SHIP/REC	Added	MANUF	

Affected Filters

Customize | Find | First 1 of 1 Last

Filter	Object ID	Tree Node	Status	Parent Node
Filter				

Tree Compare page, 2 of 2

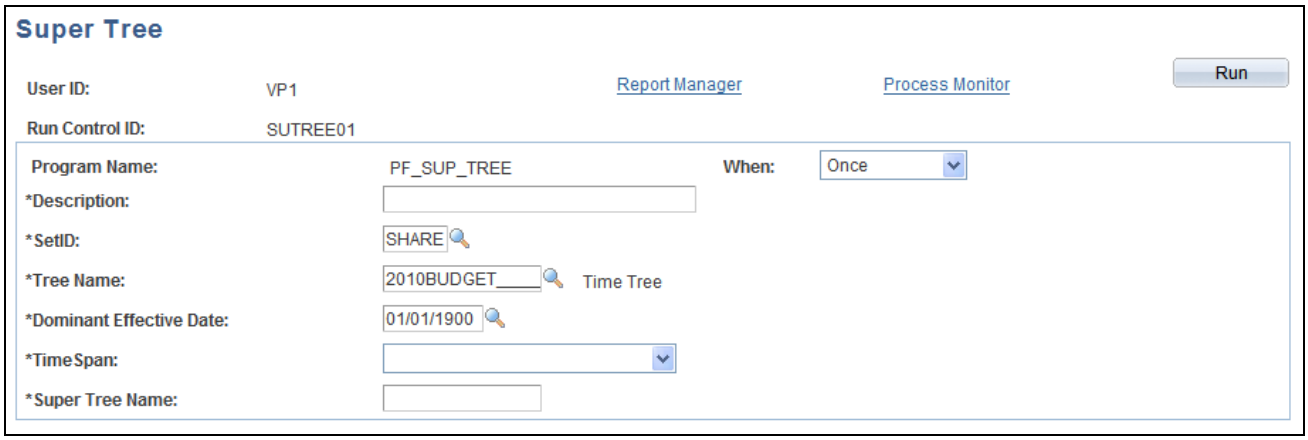
Use this page to view nodes and details (leaves) of trees that have been deleted, added, or moved from one parent to another. You may also view all metadata filters affected by the deleted nodes and details.

Filter You can change any filter by clicking on the link for that filter. The system transfers you to the Filter page on which you can modify the filter to reflect tree changes.

Note. If you determine that the tree nodes are being used in ledger to resource mappings, run the ABM Model Validation job to find the impact on the mapping rules.

Setting Up and Running Super Tree

Access the Super Tree page (EPM Foundation, Business Metadata, Tree Metadata, Update Super Tree, Super Tree).



Super Tree page

SetID SetID for the tree you want to merge.

Tree Name Name of the tree to be merged to create a super tree.

Dominant Effective Date Trees may have the same name and different effective dates. The Super Tree is created using the dominant effective date you enter in this field.

TimeSpan The timespan determines the begin and end date of the period for which the merge is performed. Trees with effective dates that are between the begin and end dates are included in the super tree.

Super Tree Name You must select a valid tree name for the super tree.

Warning! If a tree that is part of the super tree has a dead node, the dead node must be removed manually. Otherwise the super tree will not build properly. Dead nodes are created when a leaf or node is deleted. You may add a deleted node or leaf elsewhere on the tree.

Note. Any nodes or details deleted before the dominant effective date and any nodes or details added after the dominant effective date will not be included in the super tree.

Reviewing Super Tree Results

Access the Super Tree inquire page (EPM Foundation, Business Metadata, Tree Metadata, Review Super Tree, Super Tree).

Get Nodes Click this link to populate the page with the nodes of the super tree.

Get Details Click to view the related object IDs.

Setting Up and Flattening Tree Metadata

This section provides overviews of trees and tree flattening, and discusses how to:

- Define trees.
- Flatten tree metadata.

Understanding Trees

When you add trees in PeopleTools Tree Manager, you should also create a record on the Tree Metadata page to identify it to the PeopleSoft EPM system. The Tree Manager does not assign a level number to a node unless you specify that the tree either loosely or strictly enforces levels. Levels are assigned by creating a code for each level. In PeopleSoft EPM, for trees that use rules and are processed through the tree level, you must set up your trees using levels. PeopleSoft recommends specifying *strictly enforced levels* for all your trees.

PeopleSoft trees add a visual layer to show how detail items such as departments, accounts, products, channels, geography, and security fit into your organizational structure.

Trees depict hierarchical structures that represent a group of summarization rules for a particular database field. For example, a tree can specify how your manufacturing locations should be summarized, or rolled up, for reporting purposes. A tree can also represent the reporting relationships within an organization by specifying how the individual department should be summarized into territories, territories into regions, and regions into countries. Similarly, a tree can categorize items in a catalog.

The summarization rules depicted in a tree apply to the detail values of a particular field: vendors, departments, customers, or other values that you define. These detail values are summarized into nodes on a tree. The nodes may also be organized into levels to logically group nodes that represent the same type of information or level of summarization.

By building trees, you give the system a single place to look for summarization rules. Trees enable you to define rules once and then use them throughout the system. Different reports, ledgers, and security profiles might refer to parts of your company's organizational chart; therefore, all of these objects can be referenced in the same predefined tree.

For example, the values of the DEPTID field identify individual departments in your organization. You use Tree Manager to define the organizational hierarchy that specifies how each department relates to the others—departments 10700 and 10800 report to the same manager, department 20200 is part of a different division, and so on.

You create trees using the PeopleSoft Tree Manager. You can use the PeopleSoft Tree Mover to move trees between different versions of PeopleSoft databases, move tree node data, or move tree level data.

See Also

PeopleSoft Enterprise PeopleTools PeopleBook: PeopleSoft Tree Manager

Understanding Tree Flattening in the OWE

Trees are used in PeopleSoft EPM to represent hierarchies. The trees are stored in a proprietary format that is optimized for the PeopleSoft Tree Manager module. However, this format is not optimized for EPM batch processing. To convert tree structures to a format that is optimized for batch processing, PeopleSoft delivers a tree flattener process.

The Tree Flattener (PF_TREEFLT) application engine processes the hierarchy tree and converts it to a flattened format, which simplifies the structure to retrieve information such as all descendents and immediate descendents from a node with a simple join to the flattened table. The Application engine can then use very simple SQL to access the tree data that it needs. The flattened output from the Tree Flattener (PF_TREEFLT) process is loaded to a temporary table, and you can choose to load the flattened data to a permanent OWE table if you wish. The tree flattener process can flatten detail, node-oriented, and dynamic detail trees.

To flatten dynamic detail trees, the tree flattener process retrieves the dynamic detail value table name from the Tree Structure ID and inserts these values into the flattened table. During this process a static view of the tree is created as of the date and time the flattening occurs. This means that the flattener process gets the current values of the tree detail from the dynamic detail value table and uses these in the flattener table. Because of this fact, you should not use the Persist Permanently check box with dynamic detail trees unless you are sure that the dynamic table will not change.

Note. The OWE uses a different process for flattening trees than the MDW.

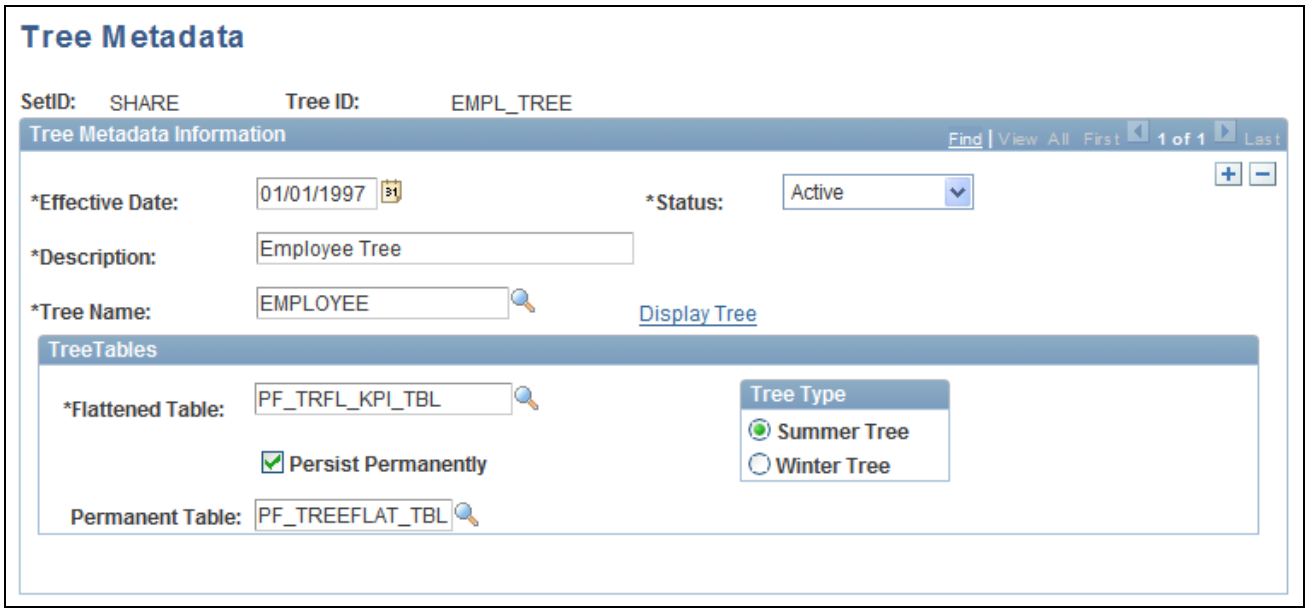
For more information on the MDW tree flattening process see the chapter *Processing Trees and Recursive Hierarchies* in your warehouse specific PeopleBook (for example, the *PeopleSoft Enterprise Campus Solutions Warehouse PeopleBook*).

Pages Used to Set Up and Flatten Tree Metadata

Page Name	*Definition Name*	*Navigation*	*Usage*
Tree Metadata	PF_METATREE_TBL1	EPM Foundation, Business Metadata, Tree Metadata, Tree Metadata	Define tree metadata.
Run Tree Flattener	PF_RUN_TREEFLAT	EPM Foundation, Business Metadata, Tree Metadata, Tree Flattener	Run the tree flattener process.
Jobstream Email Notification	PF_EMAIL_MSG	EPM Foundation, Business Metadata, Tree Metadata, Tree Flattener, Specify Email Parameters	Specify email parameters for the tree flattener notification.

Defining Tree Metadata

Access the Tree Metadata page (EPM Foundation, Business Metadata, Tree Metadata, Tree Metadata).



Tree Metadata page

Tree Metadata Information

Tree Name Select the tree for which you are creating tree metadata.

Display Tree Click to access the Tree Viewer page and view your tree.

See Chapter 16, "Setting Up and Working with Metadata for the Operational Warehouse - Enriched," Using the Tree Viewer, page 384.

Tree Tables

Flattened Table When you run the tree flattener (PF_TREEFLT) process, the preprocessor output is stored in a temporary table, which is associated with the table that you select here.

To move the preprocessor output from the temporary table to a permanent OWE table, you must select the Persist Permanently check box located on this page.

Note. You should not assign the same tree flattener table to more than one tree metadata definition.

Persist Permanently Select this check box to move the preprocessor (flattened) output to a permanent OWE table from the temporary table.

The Permanent Table field appears when you select this check box.

Note. You should not select this check box for dynamic detail trees unless you are sure that the dynamic table will not change.

Permanent Table Select a permanent OWE table you want to store the flattened output.

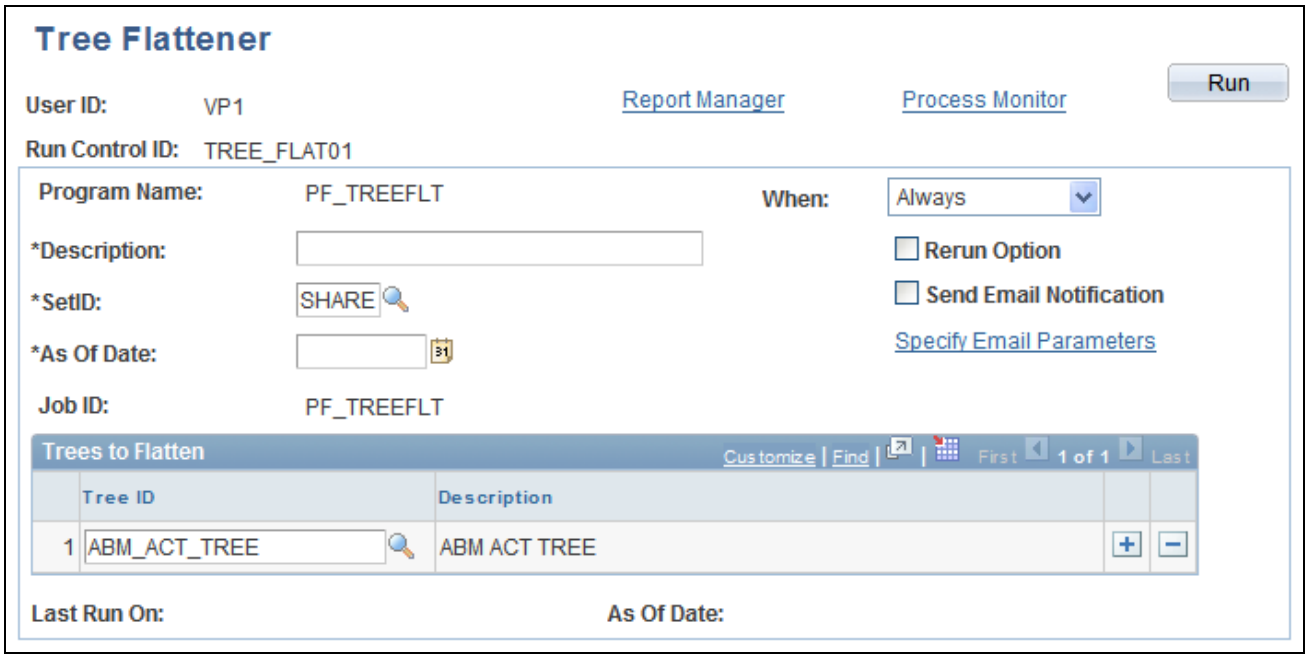
Tree Type Specify whether the tree is a summer or winter tree.

If you override a summer tree by selecting *Winter*, the tree nodes are used, and the summer tree leaf values are ignored.

You cannot override a winter tree.

Flattening Tree Metadata in the OWE

Access the Tree Flattener page (EPM Foundation, Business Metadata, Tree Metadata, Tree Flattener).



Tree Flattener page

SetID Select the setID that is associated with the tree being flattened.

Job ID Displays the job ID that is associated with the run control ID.

Because the tree flattener process is not delivered as a standalone process but as a routine that is called within application engine, you must associate the run control ID with a job ID.

Specify Email Parameters Click to access the Jobstream Email Notification page and specify email details for the email notification.

Tree ID Enter the tree ID for the tree that you want flattened

You can add rows to have multiple trees flattened.

Chapter 18

Setting Up Business Rules for the Operational Warehouse - Enriched

This chapter discusses how to:

- Set up account information.
- Specify ledger mapping defaults.
- Defining ledger event codes.
- Defining performance ledger templates.
- Define detail ledgers.
- Define ledger groups.
- Process roll-ups.
- Set up and run currency conversion

Setting Up Account Information

This section provides an overview of account information setup and discusses how to:

- Define account types.
- Define accounts.
- Define account nodes.

Understanding Accounts

Like general ledger accounts, accounts in PeopleSoft EPM hold the key to monetary values stored in ledgers. These values are held in certain buckets—accounts—according to their function.

If you have PeopleSoft General Ledger you can duplicate your general ledger account structure from your transaction system using PeopleTools Data Mover to move your accounts to your PeopleSoft EPM database. If you use another online transaction processing (OLTP) system or a legacy system, you can migrate your structure using data migration tools.

PeopleSoft EPM contains a couple of important account tables:

GL_ACCOUNT_TBL Contains all currently active general ledger and performance (PF) accounts, keyed by setID.

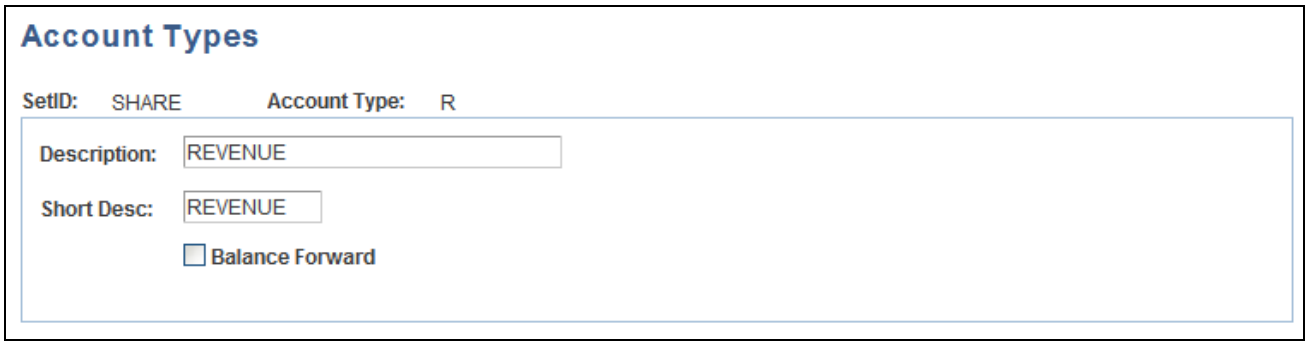
PF_ACCOUNT_VW View of the account table that filters for performance (PF) accounts only.

Pages Used to Set Up Account Information

Page Name	*Definition Name*	*Navigation*	*Usage*
Account Types	ACCT_TYPE	EPM Foundation, Business Metadata, Business Framework, Account Types	Define the account types to which you will later assign accounts. Account types describe the various categories of accounts that correspond to those on your balance sheet or income statement.
Accounts	ACCOUNT_TBL1	EPM Foundation, Business Metadata, Business Framework, Accounts	Define new PeopleSoft EPM accounts.
Account Nodes	PF_ACCT_NODE_DFN	EPM Foundation, Business Metadata, Business Framework, Account Nodes	Define the accounts nodes to which you will later assign accounts.

Defining Account Types

Access the Account Types page (EPM Foundation, Business Metadata, Business Framework, Account Types).



Account Types page

Account Type Displays a letter-code that identifies each account type. You assign account codes to new accounts (those not migrated from other systems) that you enter on the Account page.

Values include:

A: Asset

E: Expense

L: Liability

Q: Equity

R: Revenue

F: Force Balancing

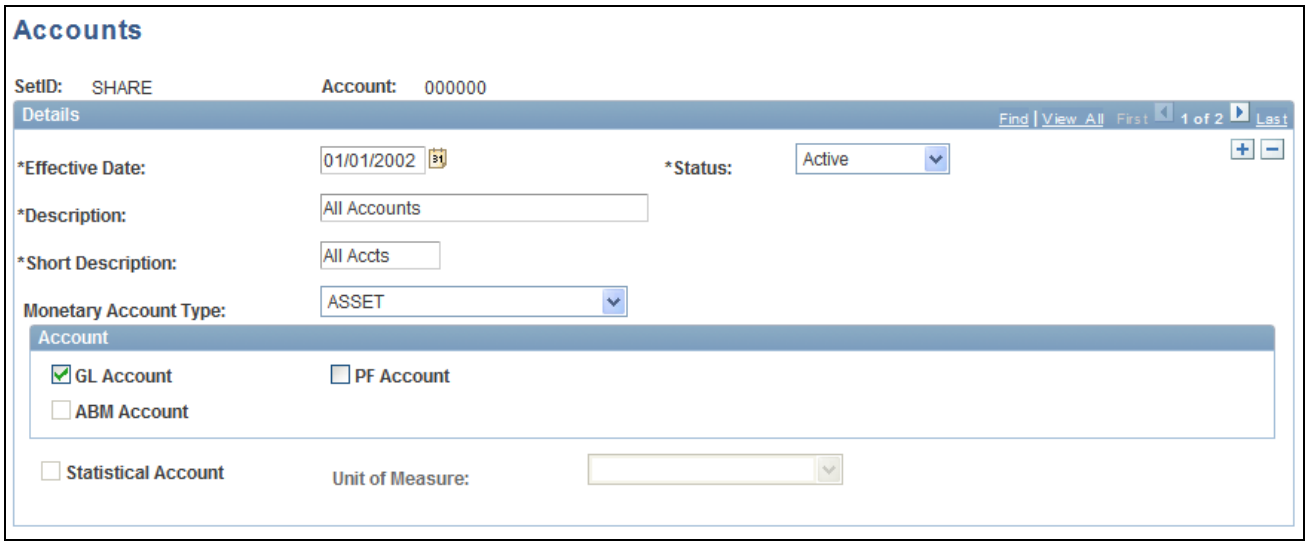
D: Detail

Balance Forward Select if you use this account type primarily to determine whether to store balance forward amounts. How you track balance forward amounts is completely up to you. For example, in your regular accounting system, you may want asset, liability, and equity accounts specified as balance forward accounts, but not revenue or expense accounts. The accounts you define later on the Accounts page will take on the balance forward attribute of the account type you assign to them.

Warning! For accounts that you use with PeopleSoft financial services industry applications it is important that you not deviate from the account type codes mentioned here. All asset accounts must be tagged with an *A*, expense accounts with an *E*, and so on.

Defining Accounts

Access the Accounts page (EPM Foundation, Business Metadata, Business Framework, Accounts).



Accounts page

Monetary Account Type Select from the account types defined on the Account Types page.

Note. To obtain account information on reports, you need to add those accounts to the system. In PeopleSoft EPM, you set up a base of general ledger accounts by migrating them from your transaction accounting system. However, when your data migration is complete, you may find that you need to add new accounts to your system.

Warning! For accounts that you will use with financial services applications—PeopleSoft Enterprise Funds Transfer Pricing, Risk-Weighted Capital, and Asset Liability Management—it is important that you not deviate from the standard account type codes. All asset accounts must be tagged with an *A*, expense accounts with an *E*, and so on.

GL Account Select if the account is a transaction-based account.

PF Account Select if the account is performance (PF) ledger.

ABM Account An account tagged as an ABM account creates more detailed information within the performance (PF) ledger. The ABM-specific fields on this page enable you to specify that the performance ledger contain both the appropriate profitability dimension (cost object), and the activity that contributed costs to the cost object.

Object Type When you select the ABM Account check box, a drop-down list box for object type appears. Select a default object type for the account.

Activity ID When you select the ABM Account check box, a drop-down list box for activity ID appears. Select a default activity for the account.

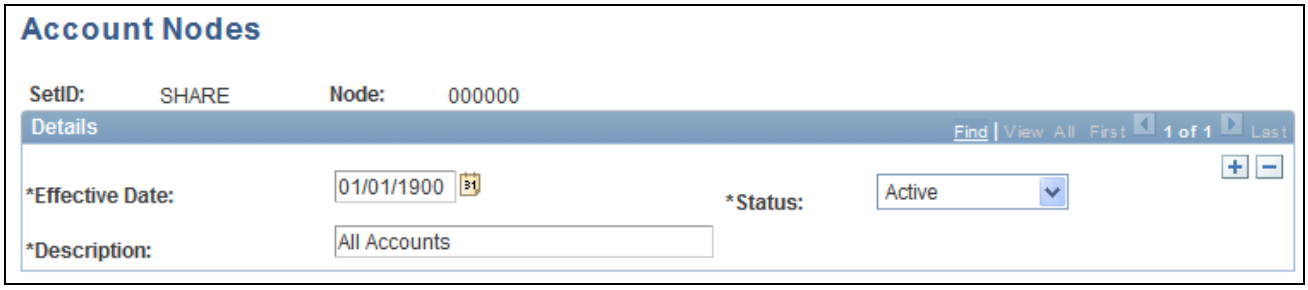
Statistical Account If you are establishing a statistical account, select the check box and select a unit of measure. Statistical accounts store only statistical amounts, not monetary amounts.

Unit of Measure If you want to track statistical amounts, select a unit measure.

Warning! Although monetary amounts are assigned to performance (PF) accounts, there is *no* reconciliation *per se,* as there is in PeopleSoft General Ledger. The performance (PF) account and source dimensions are used to store and reconcile amount facts to the same amounts stored by a general ledger account. Performance (PF) accounts can differ significantly from general ledger accounts.

Defining Account Nodes

Access the Account Nodes page (EPM Foundation, Business Metadata, Business Framework, Account Nodes).



Account Nodes page

You can define accounts nodes to which you will later assign accounts. The page displays an effective date, status, and description for each node.

Specifying Ledger Mapping Defaults

As a next step you need to specify your ledger and ChartField mapping.

On the Ledger Mapper Defaults pages, you specify the defaults and behavior for the Ledger Mapper page in EPM. The defaults you specify here determine how a new row in the Ledger Mapper page is presented and edited. You can override the defaults later on the Ledger Mapper page if you want to set up your mappings differently.

You use the Ledger Mapper to map data, such as assets and liabilities, that does not come into the system through one of the optional analytical applications. The Ledger Mapper is described later in this PeopleBook.

This section discusses how to:

- Specify tree views.
- Specify trees, default values, and validation.
- Set ledger mapping tree usage.

See Also

Chapter 21, "Setting Up and Using Profit Manager," Setting Up Ledger Mapper, page 501

Common Elements Used in This Section

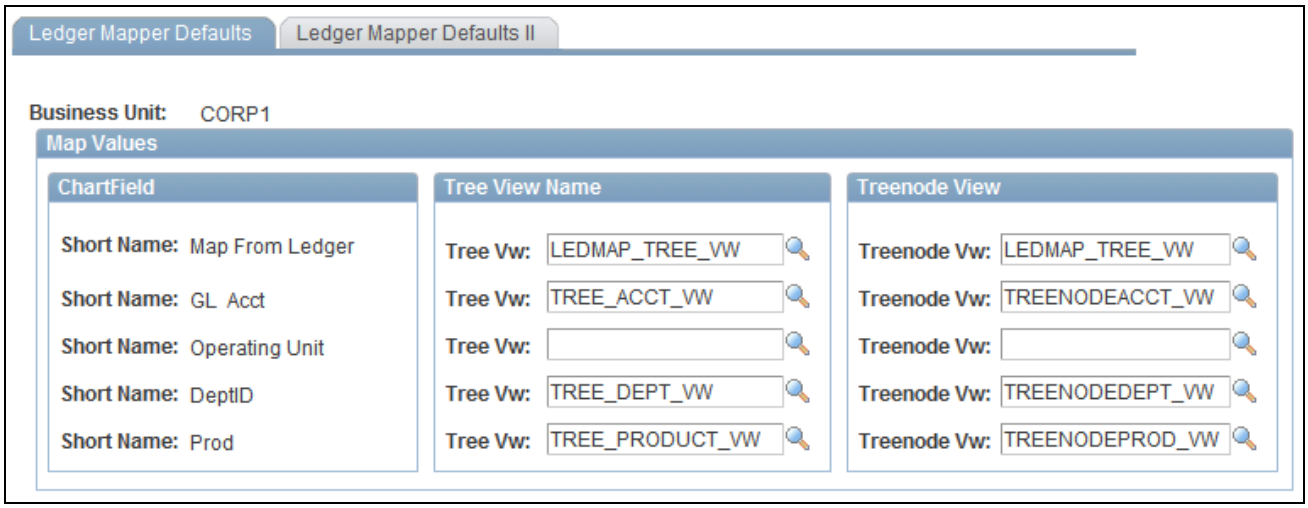
ChartField The ChartFields that appear on the list come from the LEDMAP_CHART subrecord.

Pages Used to Set Up Ledger Mapper Defaults

Page Name	*Definition Name*	*Navigation*	*Usage*
Ledger Mapper Defaults	PF_LED_TMPL_TBL1	EPM Foundation, EPM Setup, Ledger Setup, Ledgers, Ledger Mapper Defaults, Ledger Mapper Defaults	Specify tree views to filter the tree information you've set up for your ledger and ChartField mappings.
Ledger Mapper Defaults II	PF_LED_TMPL_TBL2	EPM Foundation, EPM Setup, Ledger Setup, Ledgers, Ledger Mapper Defaults, Ledger Mapper Defaults II	Specify the trees, default values, and validations that define how your ledgers and ChartFields roll up, which default values to draw from, as well as whether or not to allow blank values.
Map Ledger to Tree	PF_LM_TOPT_TBL1	EPM Foundation, EPM Setup, Ledger Setup, Ledgers, Map Ledger to Tree	Set ledger mapping tree usage.

Specifying Tree Views

Access the Ledger Mapper Defaults page (EPM Foundation, EPM Setup, Ledger Setup, Ledgers, Ledger Mapper Defaults, Ledger Mapper Defaults).



Ledger Mapper Defaults page

ChartField The page displays the ChartFields: map from ledger, general ledger account, operating unit, department ID, and product.

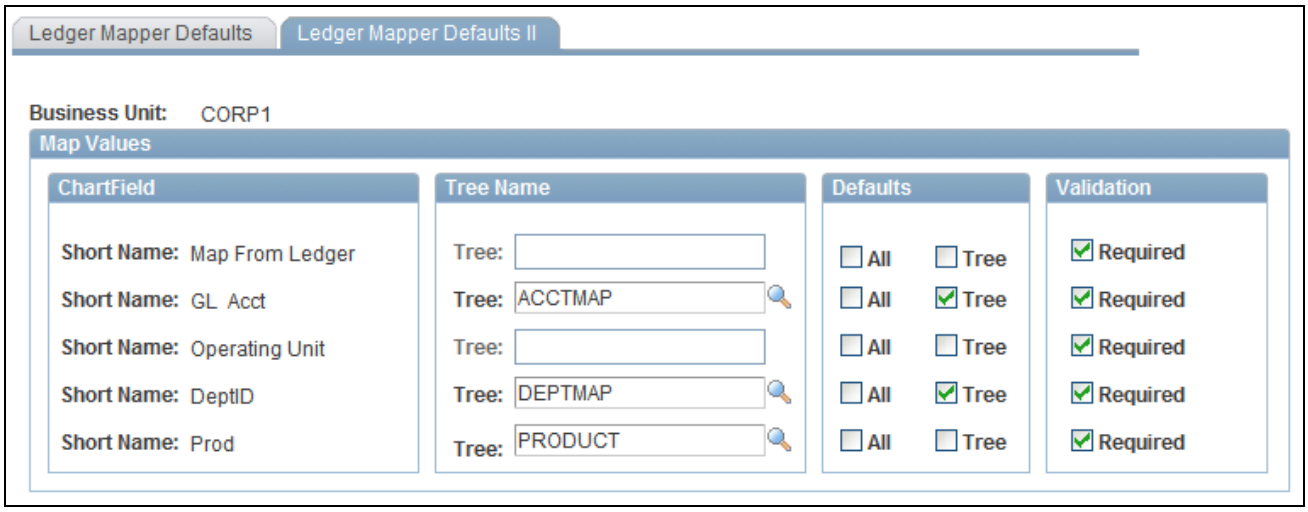
Tree View Name For each ChartField, specify the appropriate tree view name. The prompt displays all views that have Set Control Field as BUSINESS_UNIT and TREE in the name of the view.

Treenode View For each ChartField, specify the appropriate treenode view name.

Note. If you configure ChartFields of your own to add to the system, you will need to duplicate the tree view, treenode view, and prompts. The system uses the specified views to select members from the trees you specify on the Ledger Mapper Defaults II page.

Specifying Trees, Default Values, and Validation

Access the Ledger Mapper Defaults II page (EPM Foundation, EPM Setup, Ledger Setup, Ledgers, Ledger Mapper Defaults, Ledger Mapper Defaults II).



Ledger Mapper Defaults II page

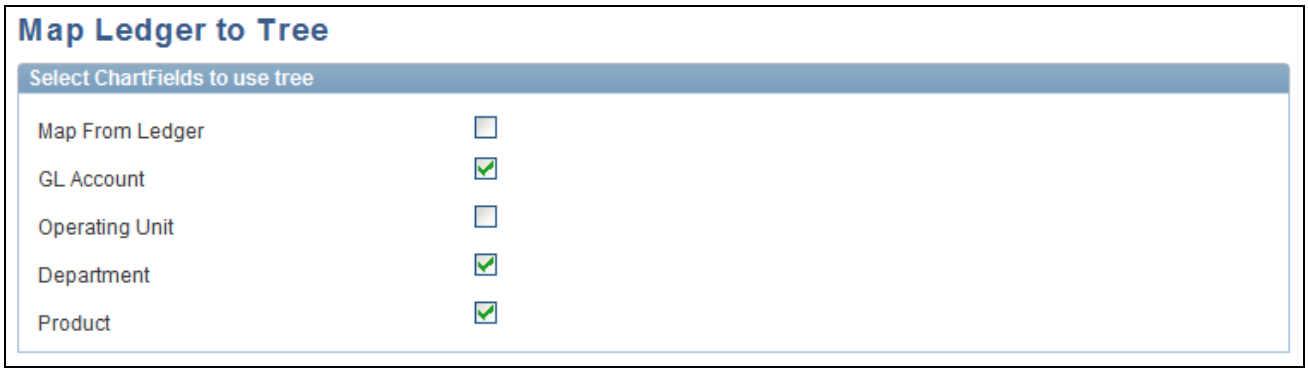
Tree Name For each ChartField including your ledger, select the appropriate tree name from which you'll map values. The tree you select should also represent how your ChartFields roll up for reporting purposes. The tree name is required if you select the Tree check box.

Defaults Select whether to map all values in the system for the ChartField row or only to those values that are filtered out through the tree and tree views you selected. If you select Tree, the system requires you to enter the tree name.

Validation To disallow the selection of blank values from prompts and drop-down list boxes for a specific ChartField or for your ledger, select the appropriate validation check box. This validation occurs on saving the Ledger Mapper Defaults component.

Mapping Ledgers to Trees

Access the Map Ledger to Tree page (EPM Foundation, EPM Setup, Ledger Setup, Ledgers, Map Ledger to Tree).



Map Ledger to Tree page

To specify whether a tree will be used while mapping the ChartField select the appropriate check box in the Select ChartFields to use trees group box.

Note. This page is used for general ledger mapping in addition to ABM ledger mapping. This page and the Ledger Mapping Defaults pages should be defined before you create ledger mapping rules.

Defining Ledger Event Codes

Ledger event codes relate performance ledger data so that you can report on the data or compare data between and within engines and models. Ledger event codes also make it possible for the Data Manager and Allocation Manager to assign an account to the target (as defined in the Data Manager or Allocation Manager) for reporting purposes and for posting.

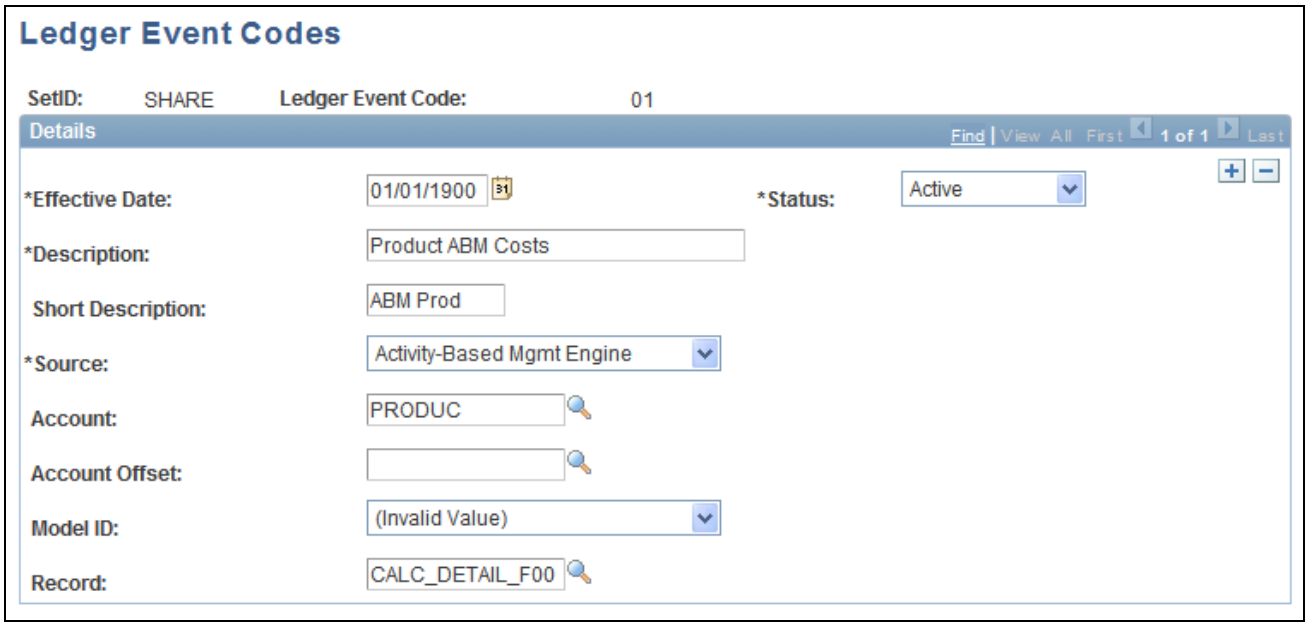
This section discusses how to establish ledger event codes.

Page Used to Set Up Ledger Event Codes

Page Name	*Definition Name*	*Navigation*	*Usage*
Ledger Event Codes	PF_SOURCE_TBL	EPM Foundation, Business Metadata, Business Framework, Ledger Event Codes	Establish ledger event codes.

Establishing Ledger Event Codes

Access the Ledger Event Codes page (EPM Foundation, Business Metadata, Business Framework, Ledger Event Codes).



Ledger Event Codes page

Source Use to specify the data that you will tie to the code. This helps you further filter where the amounts for this code originate. You can select from output that originates in your general ledger or transaction system, or in the various PeopleSoft EPM engines.

Account Specify the account that you want to assign to the ledger event code.

Account Offset You can also specify the account offset that you want to assign to the ledger event code.

Model ID (Optional) Specify a model to associate with the code.

Record (Optional) Specify a record to associate with the code.

Note. If you use PeopleSoft Enterprise Funds Transfer Pricing or Risk-Weighted Capital, you assign ledger event codes to rules through the Rules pages in those applications. For more information, see the documentation for those applications.

To use ledger event codes appropriately through the Data Manager or Allocation Manager, you need to designate the PF_LEDGER_EVENT_CD field on the TableMap and DataMap system pages.

See Also

Chapter 16, "Setting Up and Working with Metadata for the Operational Warehouse - Enriched," Setting Up Tablemaps, page 355

Chapter 16, "Setting Up and Working with Metadata for the Operational Warehouse - Enriched," Setting Up Datamaps, page 363

Defining Performance Ledger Templates

This section provides an overview of performance ledger templates and discusses how to:

- Set up a performance ledger template.
- Specify ledger template ChartFields.
- Assign temporary tables to records.

Understanding Performance Ledger Templates

A ledger template defines the physical attributes of a ledger and streamlines the ledger definition process. You define the template once and use it for all your ledgers.

When you create a ledger template, the system automatically identifies the list of ChartFields based on the ledger table selected. The ChartFields are identified based on the following criteria:

ChartField is equal to any field between the ACCOUNTING_PERIOD and CURRENCY_CD fields based on the selected ledger table's structure in the database

The currency code field is also always treated as a ChartField.

PeopleSoft provides a number of predefined ledger templates, which include default records and fields that support the ChartFields and other field and record definitions delivered with the system. (SetID does not key the template). If you want to create a new template or modify a template's records or fields, you need to follow the instructions in this section.

You define performance ledger templates using the Ledger Template component.

Note. The last page in the Ledger Template component, Budget Variables, is used by PeopleSoft Planning and Budgeting and is described in that PeopleBook.

See *PeopleSoft Enterprise Planning and Budgeting 9.1 PeopleBook*, "PeopleSoft Enterprise Planning and Budgeting Preface."

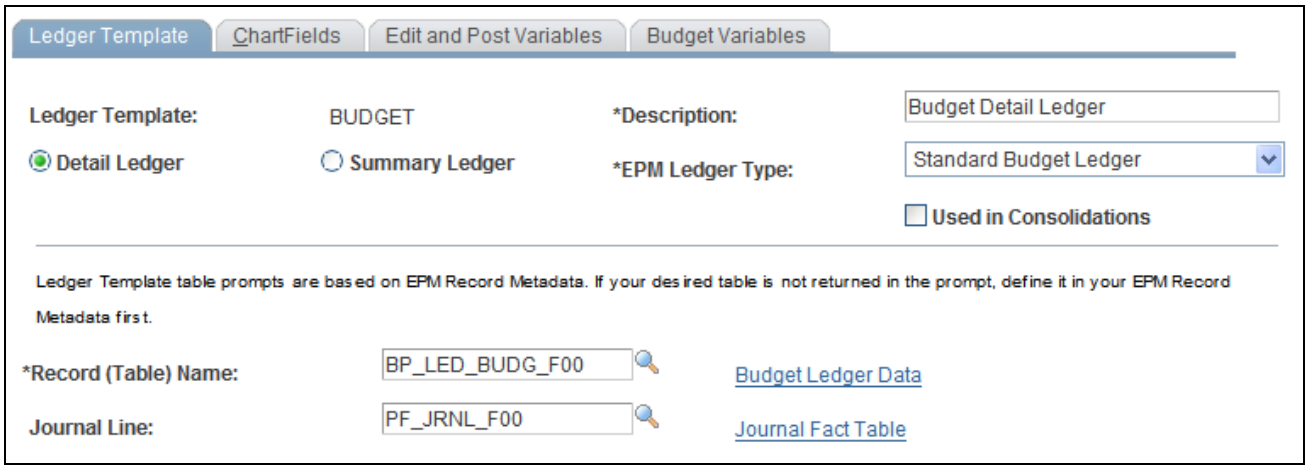
Pages Used to Define Performance Ledger Templates

Page Name	*Definition Name*	*Navigation*	*Usage*
Ledger Template - Ledger Template	LEDGER_TEMPLATE1	EPM Foundation, EPM Setup, Ledger Setup, Ledger Template, Ledger Template	Set up a performance ledger template.
Ledger Template - ChartFields	LEDGER_TMPLT_CF	EPM Foundation, EPM Setup, Ledger Setup, Ledger Template, ChartFields	Specify ledger template ChartFields.

Page Name	Definition Name	Navigation	Usage
Ledger Template - Edit and Post Variables	LEDGER_TMPLT_EDPST	EPM Foundation, EPM Setup, Ledger Setup, Ledger Template, Edit and Post Variables	Assign temporary tables to records.

Setting Up a Performance Ledger Template

Access the Ledger Template page (EPM Foundation, EPM Setup, Ledger Setup, Ledgers, Ledger Template).



Ledger Template page

Detail Select this ledger type for ledgers containing detail ledger ChartFields which serve as keys.

Summary Select this ledger type for ledgers to be used for reporting purposes.

EPM Ledger Type Unless you're working with PeopleSoft Enterprise Global Consolidations or Planning and Budgeting, you'll select *Performance Measurement Ledger*. If you change the EPM Ledger Type field, the system resets the values for the ledger records and ChartFields. Your selection here also determines what fields will be displayed on the Detail Ledger page so that the ledger data in the ledger table is uniquely identified.

Note. If you select an EPM ledger type of *Consolidation Source Ledger*, do not select a journal line table name as journal entries for these tables are not supported.

Used in Consolidations Select this check box if this ledger template is used for consolidate information for reporting.

Ledger Record Represents the ledger record that you use to store the balances for each ChartField combination that has posted activity.

Journal Line	Stores detail lines containing the monetary and statistical amounts for each journal entry.

Note. The ledger record and journal line prompts are based on PeopleSoft EPM record metadata. If your desired table is not returned in the prompt, define it in your PeopleSoft EPM record metadata first.

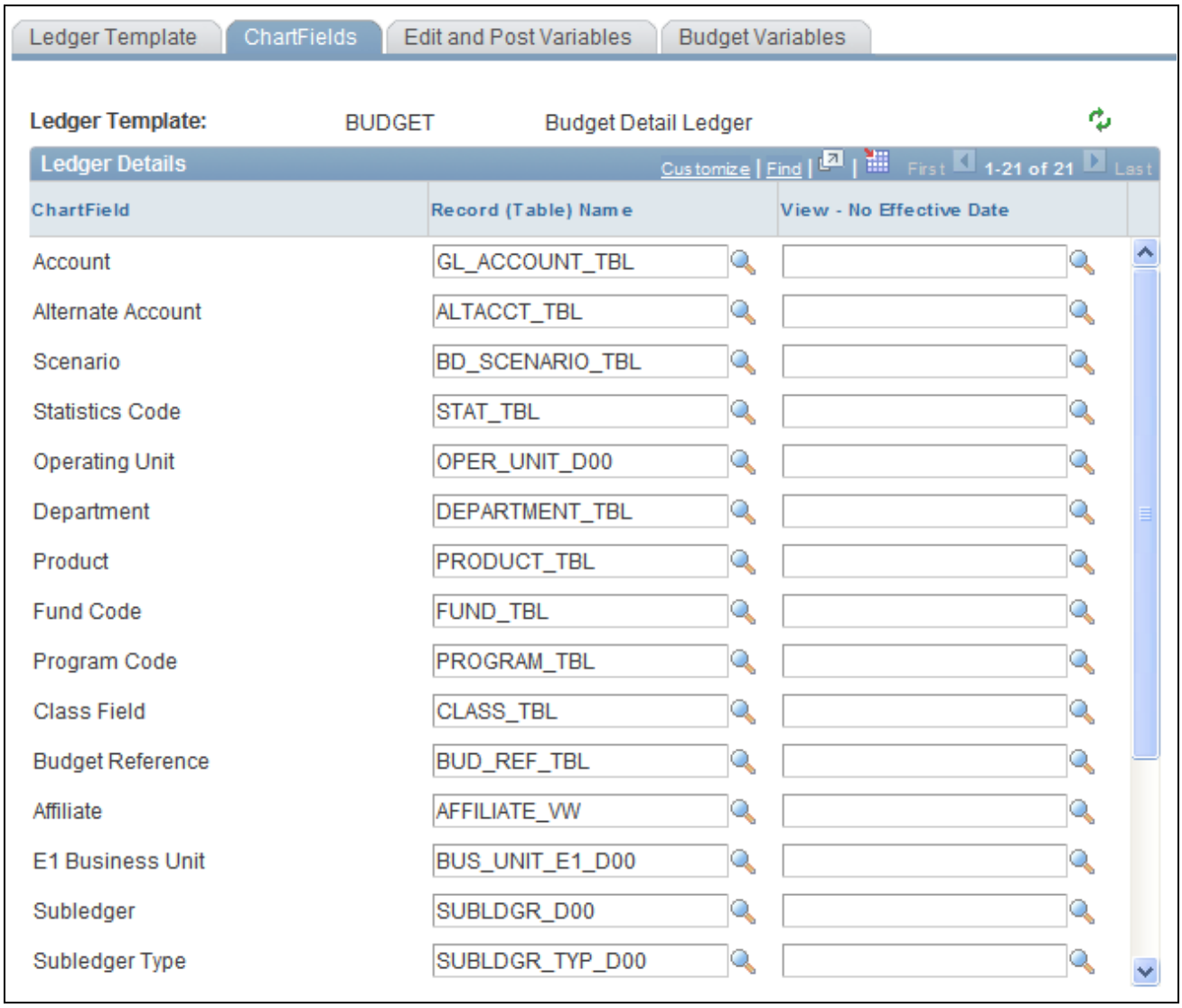
You can access the Record Metadata page for the ledger record or journal line currently selected directly from the Ledger Template page by selecting the link beside the Ledger Record and Journal Line fields. The link text varies based on the ledger record or journal line selected. In the example above, in the Ledger Record field, select the Performance Ledger link to access the Record Metadata page for the ALM_LEDGER_F00 performance ledger.

See Also

Chapter 16, "Setting Up and Working with Metadata for the Operational Warehouse - Enriched," Setting Up Record Metadata, page 348

Specifying Ledger Template ChartFields

Access the Ledger Template - ChartFields page (EPM Foundation, EPM Setup, Ledger Setup, Ledger Template, ChartFields).



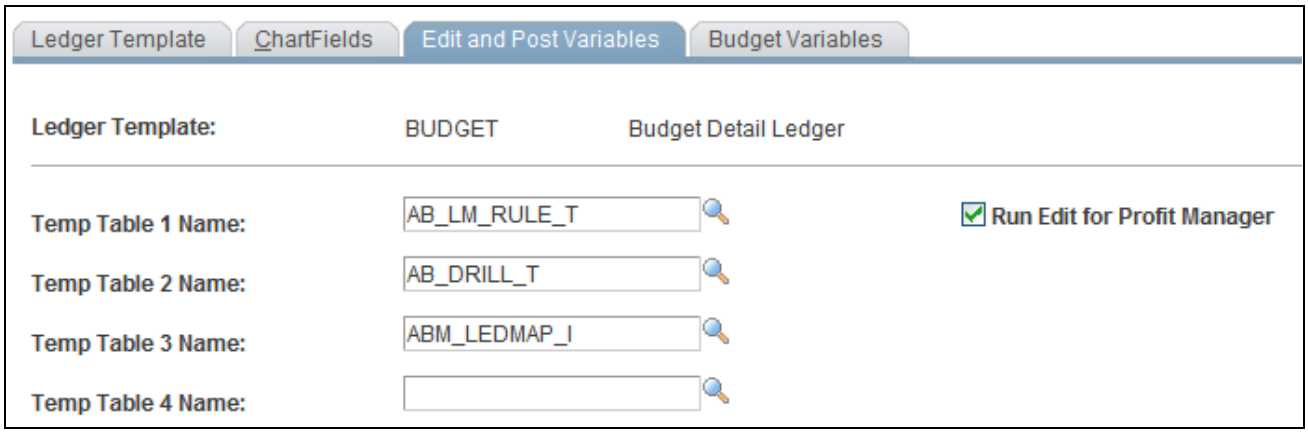
Ledger Template - ChartFields page

Select Edit Table and View - No Effective Date records to use for this template.

The page accesses a set of default edit table names and reporting views for the ChartFields in the ledger. The edit table value comes from the edit table specified in the Application Designer for this field on the ledger table. The View - No Effective Date field is used for reporting prompts when you do not want to limit the selection by effective date (because historical rows might contain ChartFields that are no longer active). You can change these values if you are modifying your system.

Assigning Temporary Tables to Records

Access the Ledger Template - Edit and Post Variables page (EPM Foundation, EPM Setup, Ledger Setup, Ledger Template, Edit and Post Variables).



Ledger Template - Edit and Post Variables page

Temp Table 1 Name, Temp Table 2 Name, Temp Table 3 Name, and **Temp Table 4 Name** — Use to assign up to four temporary tables to the ledger template. Your temporary table selection must match those tables in your ledger template. Each of the four temporary tables represent the journal and ledger records that Profit Manger uses to process your ledgers and journals.

Run Edit for Profit Manager — Select this check box to run edit for the Profit Manager.

Note. The journal and ledger tables have ChartField subrecords. If you add ChartFields, you must modify these table subrecords

---

See Also

Chapter 21, "Setting Up and Using Profit Manager," Processing and Posting Journals, page 516

Defining Detail Ledgers

Use the Detail Ledger component to link a ledger template to a detail ledger and specify the balancing options for the journals generated to a detail ledger.

Note. The second page in the Detail Ledger component, Consolidations, is used by PeopleSoft Enterprise Global Consolidations and is described in that PeopleBook.

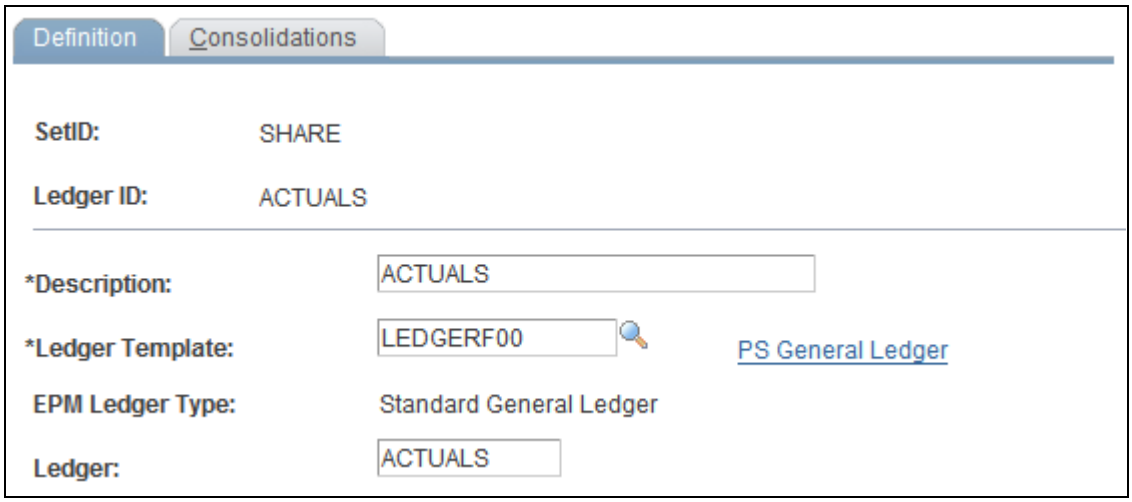
See *PeopleSoft Enterprise Global Consolidations 9.1 PeopleBook.*

Page Used to Define Detail Ledgers

Page Name	Definition Name	Navigation	Usage
Detail Ledger - Definition	LEDGER_DETAIL1	EPM Foundation, EPM Setup, Ledger Setup, Ledgers, Detail Ledger, Definition	Identify a unique set of ledger data within the ledger table.

Specifying Detail Ledgers

Access the Detail Ledger - Definition page (EPM Foundation, EPM Setup, Ledger Setup, Ledgers, Detail Ledger, Definition).



Detail Ledger - Definition page

Ledger Template Select the ledger template to link to this detail ledger.

Click the link that displays alongside the ledger template you select to access the Ledger Template page for that template.

EPM Ledger Type Displays the type of ledger selected in the Ledger Template field.

Ledger Displays for a ledger type of standard general ledger, budget project ledger, commitment budget ledger.

Budgeting Scenario Displays for a ledger type of budget project ledger and commitment budget ledger.

Scenario ID Displays for a ledger type of performance measurement ledger and consolidation ledger. Select the scenario ID to use.

Defining Ledger Groups

Ledger groups enable you to group detail ledgers by ledger template and ledger type.

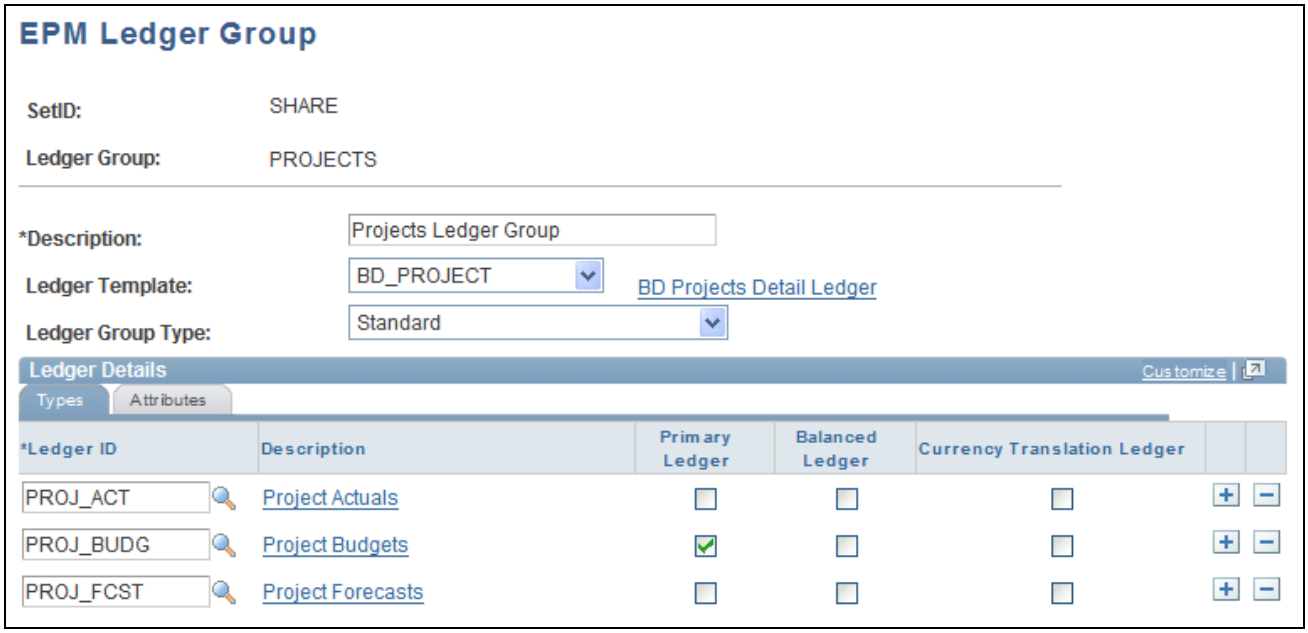
This section discusses how to set up ledger groups.

Page Used to Define Ledger Groups

Page Name	*Definition Name*	*Navigation*	*Usage*
EPM Ledger Group	LEDGER_GROUP	EPM Foundation, EPM Setup, Ledger Setup, Ledgers, Ledger Group	Set up a ledger group and assign detail ledgers to the group and identify the unique attributes of each ledger within the group.

Setting Up Ledger Groups

Access the EPM Ledger Group page (EPM Foundation, EPM Setup, Ledger Setup, Ledgers, Ledger Group).



EPM Ledger Group page

Ledger Template Select a ledger template from the drop-down list box. Ledger templates are defined on the Ledger Template pages. Click the link alongside the Ledger Template field to transfer to the Ledger Template Definition page for the selected template.

Ledger Group Type Select a group type from the following:

Average Daily Balance: Allows only a primary ledger.

Budget: Allows only a primary ledger.

Commitment Control Expense: Used by commitment control budgeting.

Commitment Control Revenue: Used by commitment control budgeting.

Standard: Allows one primary and up to nine secondary ledgers.

Translation: Allows only a primary ledger.

Entering Ledger Details

The Ledger Details grid at the bottom of the page displays two tabs: Types and Attributes.

On the Types tab select the following:

Ledger ID Select the detail ledger you want to link to this ledger group. The detail ledgers listed are those associated with the ledger template defined on the Detail Ledger page.

Primary Ledger Select this check box for the primary ledger.

You can specify one primary ledger and up to nine secondary ledgers for ledger groups that have a ledger group type of standard. For average daily balance, budget, and translation ledger group types, you can select only one ledger.

Balanced Ledger Select if this ledger contains balanced ChartFields or you want to balance the ChartFields.

Currency Translation Ledger Select if this is a currency translation ledger. Any non-primary ledger may be defined as a translation ledger. Currency translation ledgers are handled differently from normal secondary ledgers with respect to journal processing and multicurrency processing.

On the Attributes tab select the following:

Inherit Base Currency Select to inherit the base currency of the business unit. The system uses the business unit base currency in the ledger.

Base Currency If you do not elect to inherit the base currency, enter the base currency to use for the ledger.

Exchange Rate Type Default

Enter a rate type to use as a default.

This field influences which rate type will be used for the secondary lines. The calculation hierarchy is this: The system first checks for a value in rate type for the ledger (see the next field). It will always use this value for the secondary lines, regardless of what you enter in the Exchange Rate Type Default field. The system then checks for a rate type on the primary journal line. If there is one, it uses that rate type for the secondary line. If there is no value in rate type or in the primary journal line, the system uses the value entered in the Exchange Rate Type Default field.

Rate Type

Select the currency exchange rate type to use in the autogenerated journal lines for the secondary ledgers.

Note. If you're setting up a commitment control ledger group, two additional fields display: the Commitment Control Ledger Type field and the Affect Spending Authority field. These fields are specifically for commitment control purposes.

Processing Roll-Ups

This section provides an overview of EPM roll-up engines, lists common elements, and discusses how to:

- Roll up scenarios.
- Roll up calendars.
- Roll up business units.

Understanding EPM Roll-Up Engines

In EPM you can run calendar roll-ups on the application data, the performance ledger table (PF_LEDGER_F00), and roll up ledgers on the GL ledger table. In addition you can roll up scenario IDs and business units. Running the roll-up engines enables you to present the same information in a different (aggregated) way.

You can use the calendar roll-ups to roll up accounting periods based on the summary and detail calendars that you defined for your scenario. Most likely, you will use a calendar roll-up before you report on your scenario. For example, you may have monthly values in your current scenario data, but want to see quarterly values.

You can use the scenario ID and business unit roll-up engines to consolidate values. For example, when you define warehouse business units, you may specify that several business units roll up under a consolidated business unit. When you then run the business unit roll-up, the individual business unit data is aggregated into the consolidated business unit and can be reported in that way.

Note. Although the roll-up engines are usually the only job in a jobstream, they must still be set up in a jobstream. No merge is required on this engine.

Common Elements Used in This Section

Rerun

If you are rerunning this job with the same run control parameters, deselect the Rerun check box. This way, if you are assigned the same record suite and you were the last one to run in this record suite, the system does not re-resolve the tables prior to running the job. If you want to re-resolve the permanent tables, select the Rerun check box.

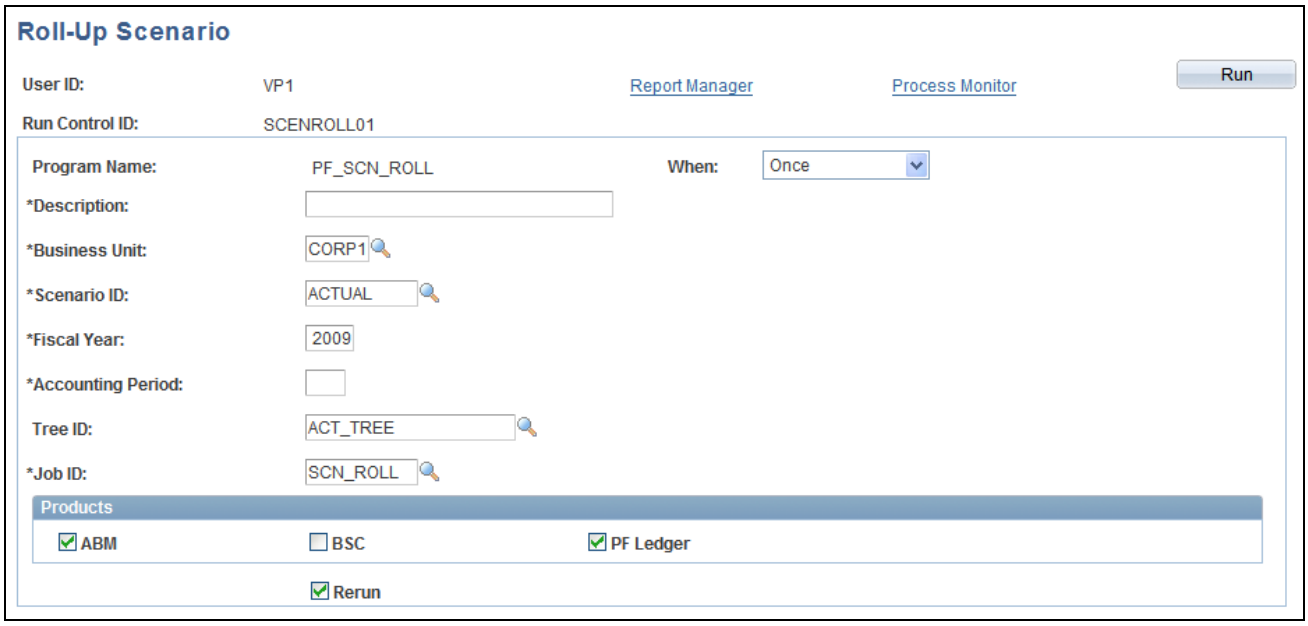
Re-resolving means that data is selected from the permanent table. If re-resolving is not needed (no new data in the permanent table for the given run control parameters), performance could be enhanced. Note that if the Rerun check box is selected, it will flatten (or reflatten) the business unit or the scenario tree.

Pages Used to Run Roll-Up Engines

Page Name	*Definition Name*	*Navigation*	*Usage*
Roll-Up Scenario	RUN_PF_SCN_ROLL	EPM Foundation, Data Enrichment Tools, Roll-up Accounting Info, Scenarios, Roll-Up Scenario	Roll up scenarios to produce an aggregated scenario set of data based on the scenario tree. Generate your reports after you run the Scenario Roll-Up engine and the Calendar Roll-Up engine.
Roll-Up Calendar	RUN_PF_CAL_RU	EPM Foundation, Data Enrichment Tools, Roll-up Accounting Info, Calendar, Roll-Up Calendar	Roll up calendars to produce an aggregate set of product data based on calendars.
Roll-Up Business Unit	RUN_PF_BU_ROLL	EPM Foundation, Data Enrichment Tools, Roll-up Accounting Info, Business Units, Roll-Up Business Unit	Roll up business units to produce an aggregate business unit set of data based on the business unit tree.

Rolling Up Scenarios

Access the Roll-Up Scenario page (EPM Foundation, Data Enrichment Tools, Roll-up Accounting Info, Scenarios, Roll-Up Scenario).



Roll-Up Scenario page

Business Unit, Scenario ID, Fiscal Year, and **Accounting Period** — Enter the business unit, scenario ID, fiscal year, and accounting period for the roll-up.

Tree ID — You can select a tree. Ensure that the tree you select contains a setID.

Job ID — Select *SCN_ROLL*.

ABM — Select this check box to roll up the scenarios in the tables associated with PeopleSoft Enterprise Activity-Based Management AB_JOB_CD_F00, CALC_DETAIL_F00, and CALC_OBJ_F00 tables.

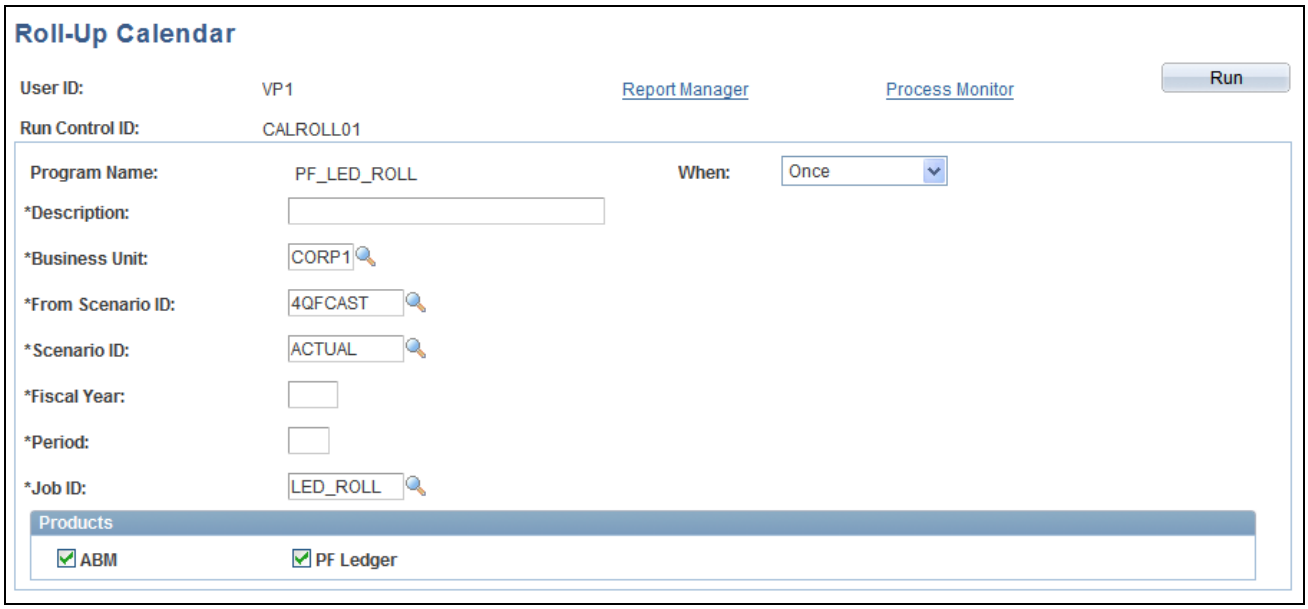
BSC — Select this check box to roll up the scenarios in the tables associated with PeopleSoft Enterprise Scorecard KP_DATAVALS_F00 table.

PF Ledger — Select this check box to roll up to the PF_LEDGER_F00 table.

Click Run to run this request.

Rolling Up Calendars

Access the Roll-Up Calendar page (EPM Foundation, Data Enrichment Tools, Roll-up Accounting Info, Calendar, Roll-Up Calendar).



Roll-Up Calendar page

Business Unit, From Scenario ID, Scenario ID, Fiscal Year and **Period** — Enter the business unit, from scenario ID, scenario ID, fiscal year, and accounting period for the roll-up.

Job ID — Select *LED_ROLL*.

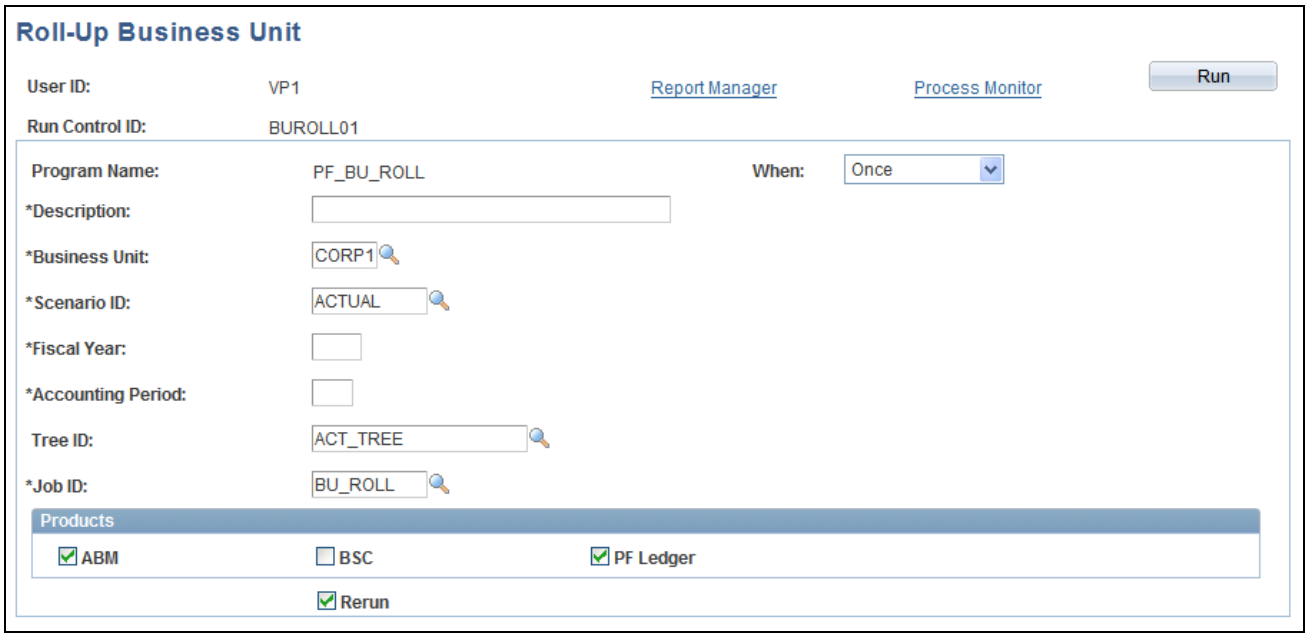
ABM — Select this check box to roll up the calendars for PeopleSoft Enterprise ABM.

PF Ledger — Select this check box to roll up performance ledger calendars.

Click Run to run this request.

Rolling Up Business Units

Access the Roll-Up Business Unit page (EPM Foundation, Data Enrichment Tools, Roll-up Accounting Info (Information), Business Units, Roll-Up Business Unit).



Roll-Up Business Unit page

Business Unit, Scenario ID, Fiscal Year and **Accounting Period** — Enter the business unit, scenario ID, fiscal year, and accounting period for the roll-up.

Tree ID — You can select a tree. Ensure that the tree you select contains a setID

Job ID — Select *BU_ROLL*.

ABM — Select this check box to roll up the business units for PeopleSoft Enterprise ABM.

BSC — Select this check box to roll up the business units associated with PeopleSoft Enterprise Scorecard.

PF Ledger — Select this check box to roll up business units for PF ledger.

Click Run to run this request.

Setting Up and Running Currency Conversion

This section provides an overview of currency conversion process for the OWE and discusses how to:

- Define currency conversion rules.
- Create currency conversion rule sets.
- Associate a conversion rule set with a job.

- Run currency conversion.
- Review multicurrency conversion messages.

See Chapter 5, "Setting Up Currency Rules for EPM," Understanding EPM Multiple Currency Processing Concepts, page 91.

Understanding Currency Conversion for the Analytical Applications

In PeopleSoft EPM, you can obtain profitability figures that include the effects of multiple currencies. The economic effects of transactions can be reconciled with accounting effects with greater accuracy. In addition, the effect of currency fluctuations (when analyzing similar business units or processes in different countries) is eliminated.

While data may be brought into EPM in many different currencies, the amounts must be converted to a single currency for each business unit in order for proper engine processing to occur. PeopleSoft EPM engines use these converted base amounts as input and output base amounts from their processes.

The Currency Conversion application engine can be used on any fact table containing the *from* and *to* currency code fields and the *from* and *to* amount fields (for example the FI_INSTR_F00 or REVENUE_F00 tables).

The Currency Conversion engine enables you to perform conversion on any fact table. However, only temporary tables defined for the fact tables are updated by the conversion engine. The Currency Conversion process should be run between two other jobs. For instance, run a job that populates the temporary table for the fact table, followed by the Currency Conversion engine process to update the temporary table, and then a job following the conversion that updates the actual fact table using the results from the temporary table.

Reports specific to currency processing are not included with PeopleSoft EPM. To view reports in a reporting currency other than your base currency, build a data mart from the PF_LEDGER_F00 table, migrate the selected data to the data mart, and run currency conversion on the data mart.

The Currency Conversion engine can be run either as a job within a jobstream, or it can be called as a function library from another PeopleSoft EPM engine.

Note. Because all ledger accounts, regardless of the engine processes, must go through the PF_JRNL_F00 before the edit and post process in order to populate to the PF_LEDGER_F00, all conversions for ledger accounts must be done on the PF_JRNL_F00 table (rather than directly on the LEDGER or PF_LEDGER tables).

Note. Any desired conversion on the LEDGER table (prior to the PeopleSoft Activity-Based Management engine using base amounts as input) must be done by the delivered ETL tool or by a configured solution created at implementation time.

Delivered Multicurrency Metadata

This table lists delivered multicurrency tablemaps, datamaps, and associated primary record names:

Tablemap/Datamap Name	*Primary Record*
MC_BALANCE	FI_IBAL_R00
MC_INSTR	FI_INSTR_F00
MC_EVENT	FI_IEVENT_R00
MC_POOLINS	FI_POOLINST_F00
MC_OPTION	FI_IOPTION_R00
MC_TRPOS	FI_TRPOS_F00
MC_PFJRNL	PF_JRNL_F00

Prerequisites

Before creating currency conversion rules and rule sets, and associating them with jobs, you must:

- Set up your currency metadata.
- Set up jobs with which you want to run OWE currency conversion.

See Also

Chapter 5, "Setting Up Currency Rules for EPM," page 91

Chapter 20, "Streamlining Processing with Jobstreams," Setting Up Job Metadata, page 475

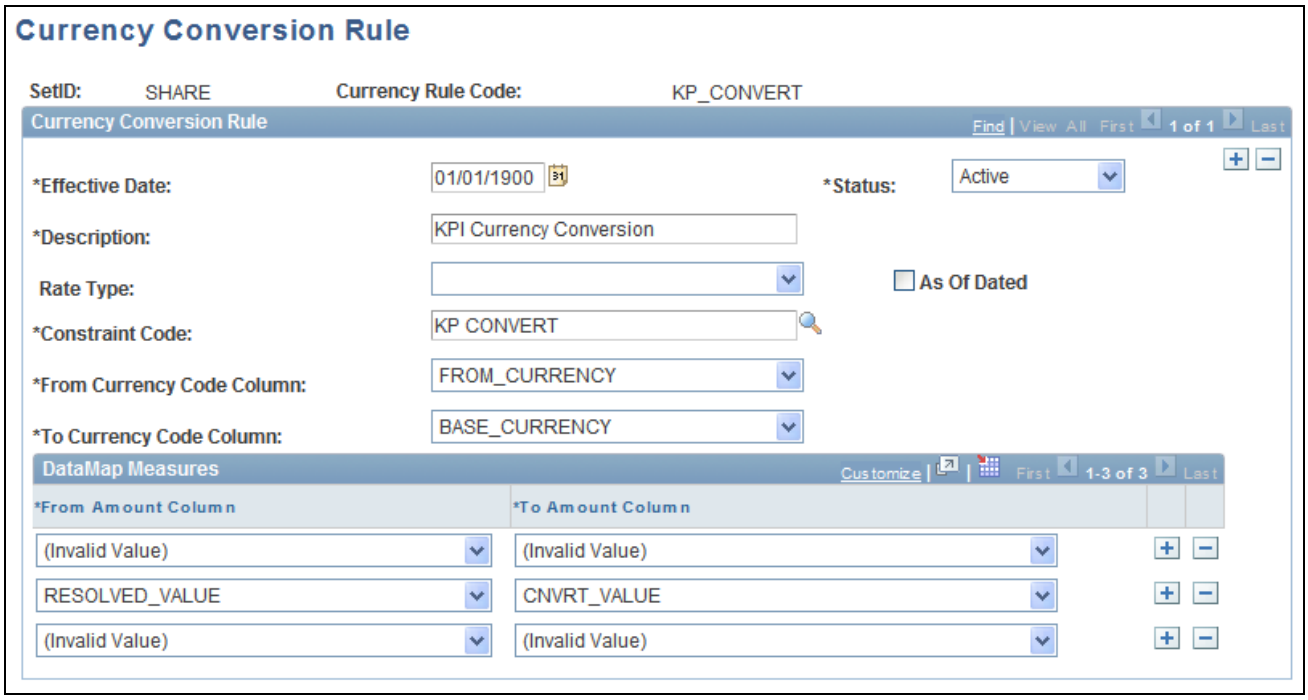
Pages Used to Set Up and Run Currency Conversion

Page Name	*Definition Name*	*Navigation*	*Usage*
Currency Conversion Rule	PF_MC_RULE_DFN1	EPM Foundation, Data Enrichment Tools, Currency Conversion, Identify Rules, Currency Conversion Rule	Describe currency conversion rules.

Page Name	Definition Name	Navigation	Usage
Currency Conversion Rule Set	PF_MC_RULESET_TBL1	EPM Foundation, Data Enrichment Tools, Currency Conversion, Create Set of Rules, Currency Conversion Rule Set	Create currency conversion rule sets by grouping one or more conversion rules , creating different rules, and, if different constraints and currency code columns are used, combining rules in a set. You must create a rule set for each rule that you want to run.
Job Conversion Rule Set	PF_MC_JOB_TBL1	EPM Foundation, Data Enrichment Tools, Currency Conversion, Associate Rule Set to Job, Job Conversion Rule Set	Associate a conversion rule set with a job. A currency conversion rule set can be assigned to more than one job, but each job can contain only one currency conversion rule set.
Message Header	PF_ENGMSG_HEAD	EPM Foundation, Data Enrichment Tools, Currency Conversion, Review Conversion Messages, Message Header	Review process information and run control parameters. **Note.** This component is used to review any engine messages that result from running a job or jobstream.
Message Detail	PF_ENGMSG_LOG	EPM Foundation, Data Enrichment Tools, Currency Conversion, Review Conversion Messages, Message Detail	Review engine message details.

Describing Currency Conversion Rules

Access the Currency Conversion Rule page (EPM Foundation, Data Enrichment Tools, Currency Conversion, Identify Rules, Currency Conversion Rule).



Currency Conversion Rule page

Rate Type Select the exchange rate type to use for this rule. The type that you specify overrides any rate type specified by the business unit. If you don't select a rate type, the system uses the rate type specified for the business unit.

As of Dated Select to use the as of date. If you do not select this check box, the system uses the period end date (based on the jobstream run parameters) as the effective date for the exchange rate. If you select this check box, any job running with this currency conversion rule uses the as of date to determine the exchange rate.

Constraint Code Specify a constraint code. Constraints are defined using the Constraint component.

From Currency Code Column Select the appropriate column from your constraint code. Any field that is *not* defined in the constraint as a measure is a valid value.

To Currency Code Column Select the appropriate column from your constraint code. Any field that is *not* defined in the constraint as a measure is a valid value.

From Amount Column Select the appropriate column. Any field defined in the constraint as a measure, and that has 1 ? 3 decimal places, is a valid value.

To Amount Column Select the appropriate column. Any field defined in the constraint as a measure, and that has 1 ? 3 decimal places, is a valid value.

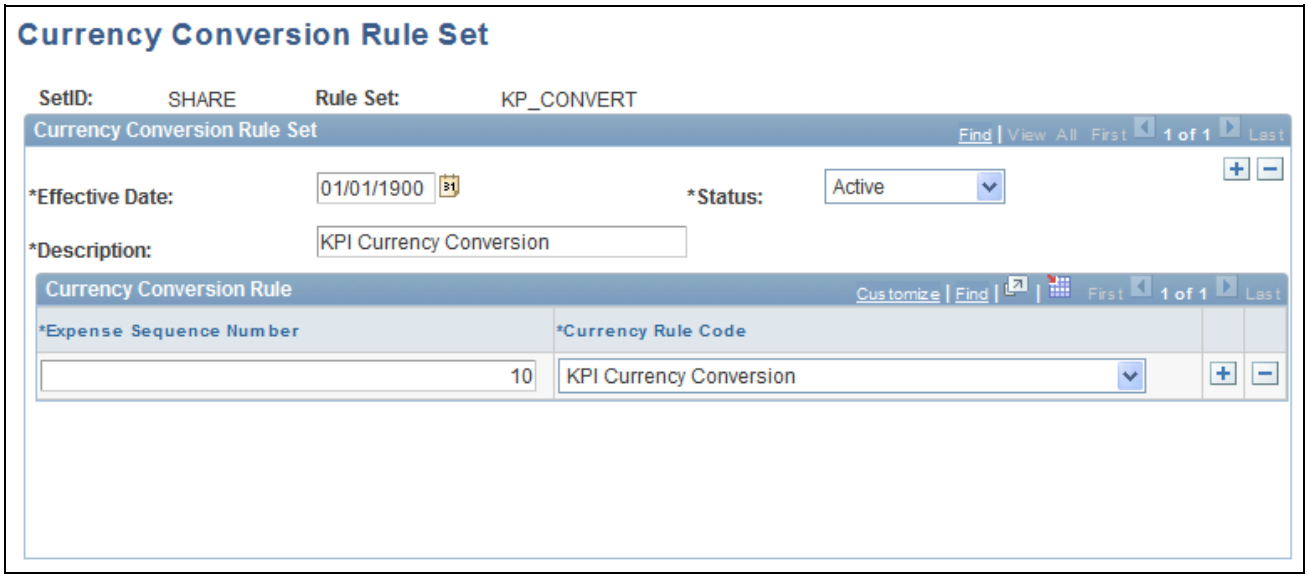
Note. If your amount columns have more than four decimal places, you must define a view for this table to redefine your amount fields. You then use this view to build your metadata and ultimately the constraint that is used as input to the multicurrency rule.

You can add more rows. You must enter field names for at least one *from* and one *to* currency code and at least one *from* and one *to* amount column. You can enter multiple *from* and *to* amount column field names for constraints that contain multiple amount fields, such as the Instrument table (FI_INSTR_F00). After you have set up currency conversion rules, you can create a conversion rule set.

Note. For each rule that you create, you must establish multicurrency tablemaps (identified as target tables), multicurrency datamaps (with amounts specified as measures), and multicurrency constraints. PeopleSoft EPM is delivered with standard table and datamaps for multicurrency. They can be identified by the MC_ prefix.

Creating Currency Conversion Rule Sets

Access the Currency Conversion Rule Set page (EPM Foundation, Data Enrichment Tools, Currency Conversion, Create Set of Rules, Currency Conversion Rule Set).



Currency Conversion Rule Set page

Expense Sequence Number Enter a number to control the order in which the Currency Conversion engine runs your rules. To accommodate the addition of future currency rules, use numbers such as 100, 200, 300.

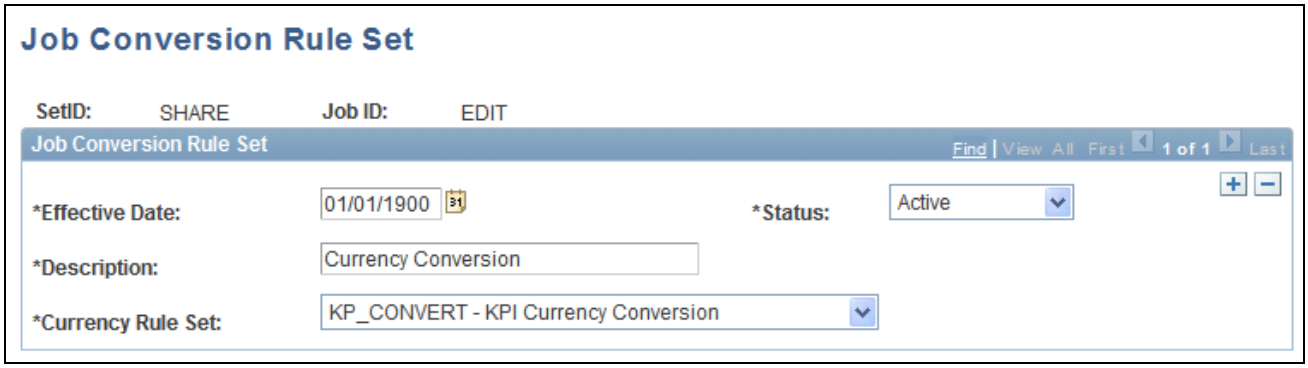
Currency Rule Code Select the name of the rule that you created.

Note. Each currency rule must be in a rule set. You can also use the rule set to combine multiple currency rules.

After you have set up the currency conversion rule set, you need to associate the rule set with a job.

Associating a Conversion Rule Set with a Job

Access the Job Conversion Rule Set page (EPM Foundation, Data Enrichment Tools, Currency Conversion, Associate Rule Set to Job, Job Conversion Rule Set).



Job Conversion Rule Set page

Job ID Enter the job with which you want to associate this rule set. You create jobs on the Job Metadata page. When you run this job in a jobstream, the system processes the currency conversion rules based on the rule set associated with the job.

Currency Rule Set Select the rule set that you created. After you have created a job rule set, you can run the Currency Conversion engine using the rules you have set up.

Note. A currency conversion rule set can be assigned to more than one job, but each job can contain only one currency conversion rule set.

Running the Currency Conversion Process

After you have completed this setup and set up your jobstreams, the Currency Conversion engine runs as part of the jobs that you defined. The engine ensures that all transactions are coded with the information that the system needs to manage multiple currency transactions. The timing of running the Currency Conversion process varies based on your business processes and the PeopleSoft EPM engines involved in your processes. Because most engines use the base currency amounts in their calculations, you must run the Currency Conversion process prior to the other engine processes for the proper base currency amounts to be available for engine processing.

The Currency Conversion engine affects any record in the warehouse that contains currency code and amount fields.

See Also

Chapter 18, "Setting Up Business Rules for the Operational Warehouse - Enriched," Delivered Multicurrency Metadata, page 444

Reviewing Multicurrency Conversion Messages

Access the Message Header (EPM Foundation, Data Enrichment Tools, Currency Conversion, Review Conversion Messages, Message Header) and Message Detail pages (EPM Foundation, Data Enrichment Tools, Currency Conversion, Review Conversion Messages, Message Detail).

Consider the following when you review dependencies:

- All amount fields for a record must be migrated to the warehouse in a single currency.

  For example, you cannot have instrument balance in USD and payment amounts in MXP.

- Multicurrency tablemaps must be designated as *target* tablemaps.
- Multicurrency datamaps must use the multicurrency tablemaps.
- Amount fields on multicurrency datamaps must be designated as *measures*, because the prompt table for these amount fields on the rule page uses a view that filters out anything that is not designated as a measure.
- Any constraints used on the rules must be specific to the multicurrency datamaps.
- Jobs that are to be run using the Currency Conversion engine must have one conversion rule set defined, and each rule set must contain at least one rule.

Note. These pages are described in detail in another chapter of this PeopleBook.

See Chapter 20, "Streamlining Processing with Jobstreams," page 461.

Chapter 19

Setting Up Models and Scenarios

This chapter provides an overview of models and scenarios and discusses how to define models and scenarios.

Understanding Models and Scenarios

Models enable you to define how to measure performance within your organization and the types of data that you want to review through various reports. For example, you may want to review revenue information on a region-by-region basis—a very high-level scope. Or, you may want to review only those activities that relate to a certain product line for certain types of resources—a very narrow scope. Each analytical application uses models and scenarios differently.

Regardless of the size or scope of your model, you must *initially* proceed with the same set of steps. Likewise, when you set up models you perform setup steps that are common to all products in the PeopleSoft EPM product line. However, how you complete the process depends on which EPM products you license.

Note. You should refer to your application-specific PeopleBook for more information on models and scenarios in your product.

Modeling Terminology

When using models, you should familiarize yourself with the following terms:

- Parent model: The master model that is used as the basis for an impact study.
- Child model: A child model stores the changes to the master or parent model.

  The child model inherits all the properties from the parent model.
- Scenario: You create a scenario ID for each parent and child model that you want to study.

  This scenario ID is used on all run controls.

Object-Based Modeling

Object-based modeling enables you to simulate various changes in an organizational model and study the impact on costs, revenue, and profitability. Instead of copying the whole model, you can copy only the data that needs to be changed. This reduces the volume of data copied and helps to identify the changes made and any effects the changes may have. Object-based modeling helps to define just the changes for the child model. All the unchanged data for the child model is inherited from the parent model. Enhanced modeling features improve the maintenance, reusability, and flexibility of a model, enabling you to create parent and child models easily.

Object-based modeling can be used not only in PeopleSoft Activity-Based Management but also in other applications in PeopleSoft EPM so that you can use your system to do planning and simulation. Scenarios for planning and simulation can be defined as child models. Child models represent the business decisions and assumptions for the scenarios. Scenarios can be simulated for each child model using various forecast distributions for a given timespan. Results for various scenarios can be compared to select the best case scenario.

Scenarios

Once you establish warehouse business units and setIDs, you create model IDs (where you define the parent and child relationships) and then point to those model ID by means of a scenario ID. Scenarios:

- Serve as a wrapper to run all analytic models.

  When you run the analytical application engines, the engines tie the models together by means of the scenario ID.

- Enable you to work with what-if scenarios by creating more than one scenario with different models attached for what-if comparisons.

There are two types of scenarios:

- Historical.
- Forecast.

To set up scenarios:

1. Complete the Scenario Definition page.
2. Select the type of scenario (historical or forecast) and complete the appropriate pages.
3. Specify the rates that you want to use on the Economic Assumptions page.
4. Assign the scenarios to a business unit using the PF Unit Scenario Definition page.

Defining Models and Scenarios

This section discusses how to:

- Create model definitions.
- Set up historical scenarios.

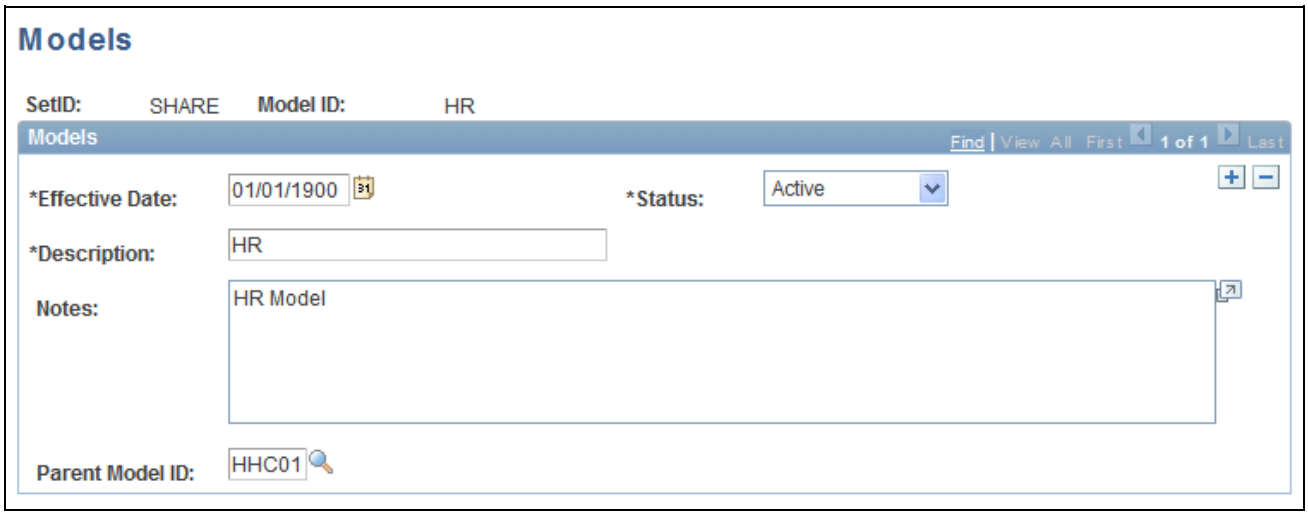
- Enter historical scenario business rules.
- Enter historical scenario economic assumptions.
- Set up forecast scenarios.
- Enter forecast scenario business rules.
- Enter forecast scenario economic assumptions.
- Assign scenarios to business units.

Pages Used to Define Models and Scenarios

Page Name	*Definition Name*	*Navigation*	*Usage*
Models	PF_MODEL_TBL1	EPM Foundation, Business Metadata, Business Framework, Models	Create a model definition.
Scenarios - Definition	PF_SCENARIO_DFN1	EPM Foundation, Business Metadata, Business Framework, Scenarios, Definition	Create a scenario definition.
Scenarios - Historical Business Rules	PF_SCENARIO_DFN2	EPM Foundation, Business Metadata, Business Framework, Scenarios, Historical Business Rules	For an historical scenario, link to a default model ID for the scenario.
Scenarios - Forecast Business Rules	PF_SCENARIO_DFN5	EPM Foundation, Business Metadata, Business Framework, Scenarios, Forecast Business Rules	For a forecast scenario, specify a fiscal year and accounting period for all model IDs that you want to use.
Scenarios - Economic Assumptions	PF_SCENARIO_DFN3	EPM Foundation, Business Metadata, Business Framework, Scenarios, Economic Assumptions	Specify the rate type for your scenario.
Scenarios - Notes	PF_SCENARIO_DFN6	EPM Foundation, Business Metadata, Business Framework, Scenarios, Notes	Describe the scenario's purpose in greater detail.
Warehouse Business Unit Scenario Definition	PF_BU_SCENARIO_DFN	EPM Foundation, Business Metadata, Business Framework, WBU Scenario Definition, Warehouse Business Unit Scenario Definition	Assign the scenarios you have defined to the appropriate business units.

Creating Model Definitions

Access the Models page (EPM Foundation, Business Metadata, Business Framework, Models).

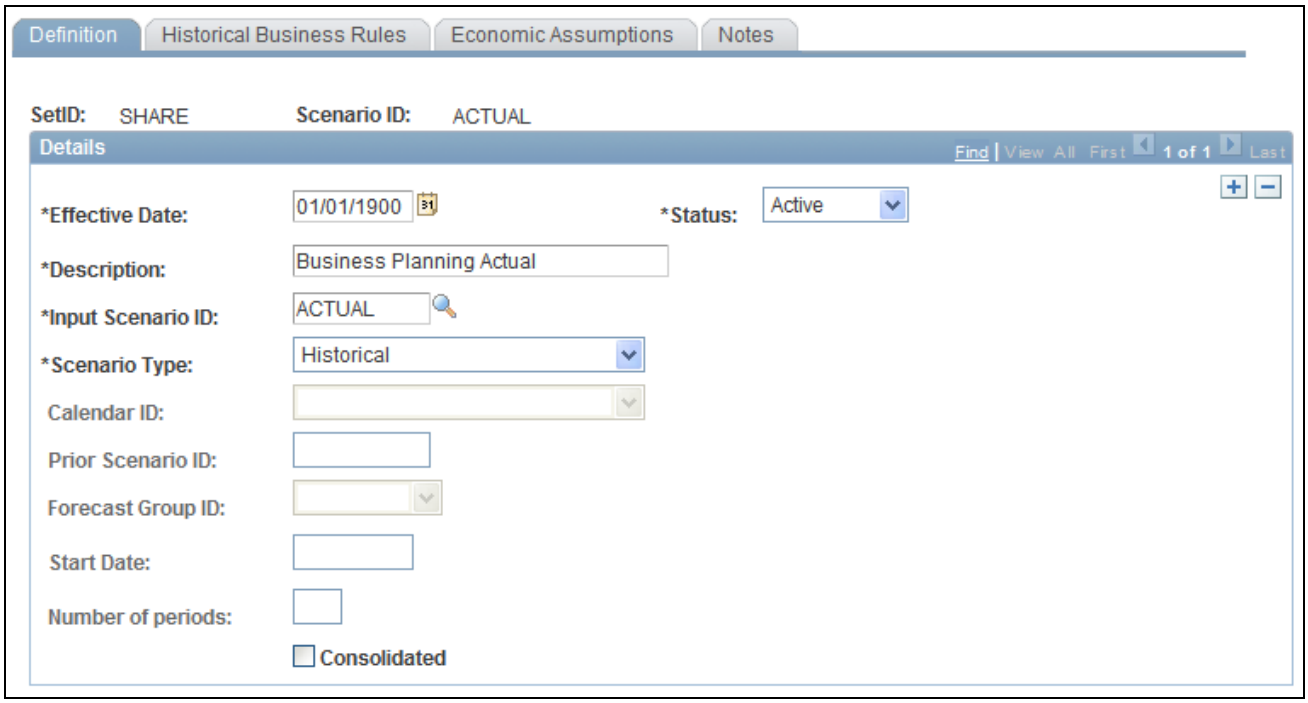


Models page

Parent Model ID For a child model that rolls up to another model, specify that ID of the parent model in this field.

Setting Up Historical Scenarios

Access the Scenarios - Definition page (EPM Foundation, Business Metadata, Business Framework, Scenarios, Definition).



Scenario - Definition page

Input Scenario ID By default, the input scenario ID is the same as the scenario ID. When you run a jobstream, Resolver uses the input scenario ID associated with the run control scenario ID to select data from a scenario-keyed table. This way, you can access data from a scenario other than the one you provide at run time.

Scenario Type Select the *Historical* scenario type.

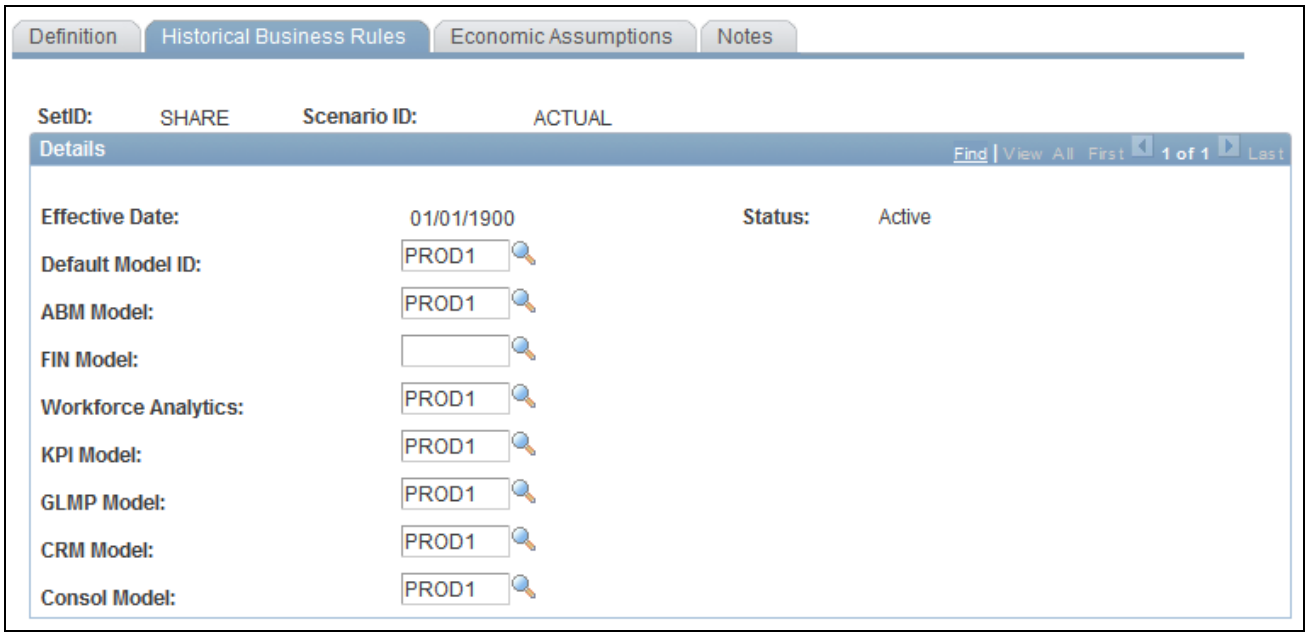
The component displays the Historical Business Rules tab.

Note. Depending on the scenario type you select, certain fields appear and others are hidden on all tabs within this component.

Consolidated Select this check box to identify this scenario as one to which other scenarios will roll up.

Entering Historical Scenario Business Rules

Access the Scenarios - Historical Business Rules page (EPM Foundation, Business Metadata, Business Framework, Scenarios, Historical Business Rules).



Scenarios - Historical Business Rules page

Default Model ID If you specify a default model ID, then the rest of the models listed default to that ID. Otherwise you can select specific models for the various applications

ABM Model Model ID to use for your PeopleSoft Activity-Based Management application.

FIN Model Model ID to use for financial analytics.

Workforce Analytics Model ID to use for your Workforce Analytics applications.

KPI Model Model ID to use for Key Performance Indications and scorecards.

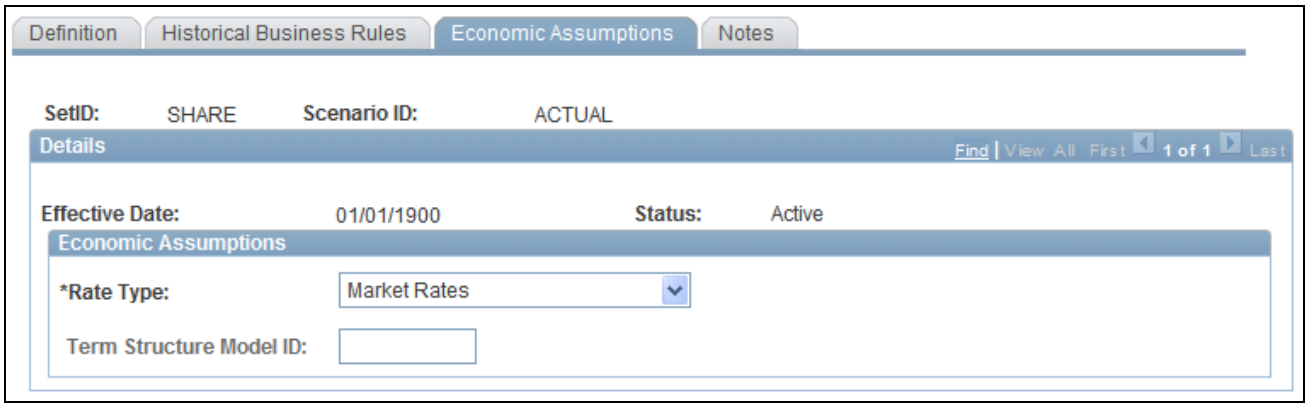
GLMP Model Model ID for the ledger mapper.

CRM Model Model ID to use for your CRM analytics applications.

Console Model Model ID to use for consolidations.

Entering Historical Scenario Economic Assumptions

Access the Scenarios - Economic Assumptions page (EPM Foundation, Business Metadata, Business Framework, Scenarios, Economic Assumptions).



Scenarios - Economic Assumptions page

Rate Type

Select from the following:

Market Rates: Use current interest rate information.

Deterministic Rates: Enables you to select a hypothetical interest rate environment previously created within PeopleSoft Asset Liability Management. Deterministic rates play an integral role in modeling a financial services institution's exposure to interest rate risk. They enable you to model dynamic interest rate environments explicitly. If you define a scenario with either a deterministic rate assumption or a stochastic one, then you must select the term structure model ID to accompany the scenario.

Stochastic Rates: Scenarios with this rate type are used to randomly generate interest rate scenarios, and they are used in risk management. If define a scenario with either a deterministic rate assumption or a stochastic one, then you must select the term structure ID to accompany the scenario.

Age-Graded Table: Used for PeopleSoft Enterprise Workforce Analytics. The age-graded table includes effective dates of rate and rating factors (such as age ranges, sex, and smoking status) by employer and employee. The arrays of values used to calculate the cost of a plan to an employee. Rates can be age-graded, service-related, or general, depending upon the benefit plan type.

Flat Rate Table: Contains information on where you define rates to be charged per selected frequency for a particular benefit program or plan. The Flat Rate table represents a constant value. For all subsequent years, as well as the first time it is done, flat rate is calculated simply by applying the flat rate percentage to the average balance and allocating this amount among the periods.

Salary Percent Table: Used in PeopleSoft Enterprise Planning and Budgeting, this table contains information related to earnings as a percentage of the base or gross salary amount.

Service Rate Table: Used in PeopleSoft Enterprise Workforce Analytics, this table stores information such as service rate ID, effective date, pay frequency, rate per unit, service intervals, total rate, employer portion, and employee portion. Service rates are employee related rates. For example, a service rate would be an employee's days of vacation accrued per year or per month for a given vacation plan.

Note. You create term structure model IDs as part of your setup for using PeopleSoft FSI applications.

Setting Up Forecast Scenarios

Access the Scenarios - Definition page (EPM Foundation, Business Metadata, Business Framework, Scenarios, Definition).

Input Scenario ID By default, the input scenario ID is the same as the scenario ID. When you run a jobstream, Resolver uses the input scenario ID associated with the run control scenario ID to select data from a scenario-keyed table. This way, you can access data from a scenario other than the one you provide at run time.

Scenario Type Select the *Forecast* scenario type. The component displays the Forecast Business Rules tab.

Note. Depending on the scenario type you select, certain fields appear and others are hidden on all tabs within this component.

Calendar ID Select the appropriate calendar ID. The calendar ID is used as a basis for the Prior Scenario ID and Forecast Group ID fields.

Note. There is a relationship between the calendar ID on this page and the calendar ID on the PF Unit Scenario Definition page. If an entry has been made in the PF Unit Scenario Definition page that associates a calendar ID to a scenario and business unit combination, you cannot update the Scenario Definition page calendar ID (it is disabled). If the Scenario Definition Calendar ID needs to be changed, the corresponding row for the scenario must be deleted from the PF Unit Scenario Definition page first. This will enable the Calendar ID field on the Scenario Definition page.

Prior Scenario ID Select the appropriate prior scenario ID. Only previously created scenario IDs with the same calendar ID are available for selection. The system uses this scenario ID as input to PeopleSoft Activity-Based Management. A prior scenario ID is required if you are specifying a model for the Financial Statement Simulator on the Forecast Business Rules page.

Forecast Group ID Select the appropriate forecast group ID for the scenario. Only previously created forecast group IDs with the same calendar ID are available. Forecast group IDs are defined in the PeopleSoft Analytic Forecasting module.

Start Date Enter the start date for the scenario. For example, if you specify a monthly calendar with a number of periods equal to 2 and specify a start date of July 12, 2002, then the system populates the next page with periods 7 and 8.

Number of Periods Enter the number of periods related to your start date. This defines the dates in the grid on the Forecast Business Rules page.

Consolidated Select this check box to identify this scenario as one that other scenarios drill up to.

Entering Forecast Scenario Business Rules

Access the Scenarios - Forecast Business Rules page (EPM Foundation, Business Metadata, Business Framework, Scenarios, Forecast Business Rules).

Enable Calendar Fill In This is selected by default. The forecast scenario creates a start date on the exact date specified within the first period. Depending on your selections, there could be time overlap or gaps. Calendar fill in helps avoid these gaps. Calendar fill in adds model IDs to blank rows for the grid lines not selected.

Specify the model IDs you want to use for each year and period. The selections you make depend on the analytical applications you have installed.

Note. You must specify a prior scenario ID on the Scenario Definition page to select a model of Financial Statement Simulator.

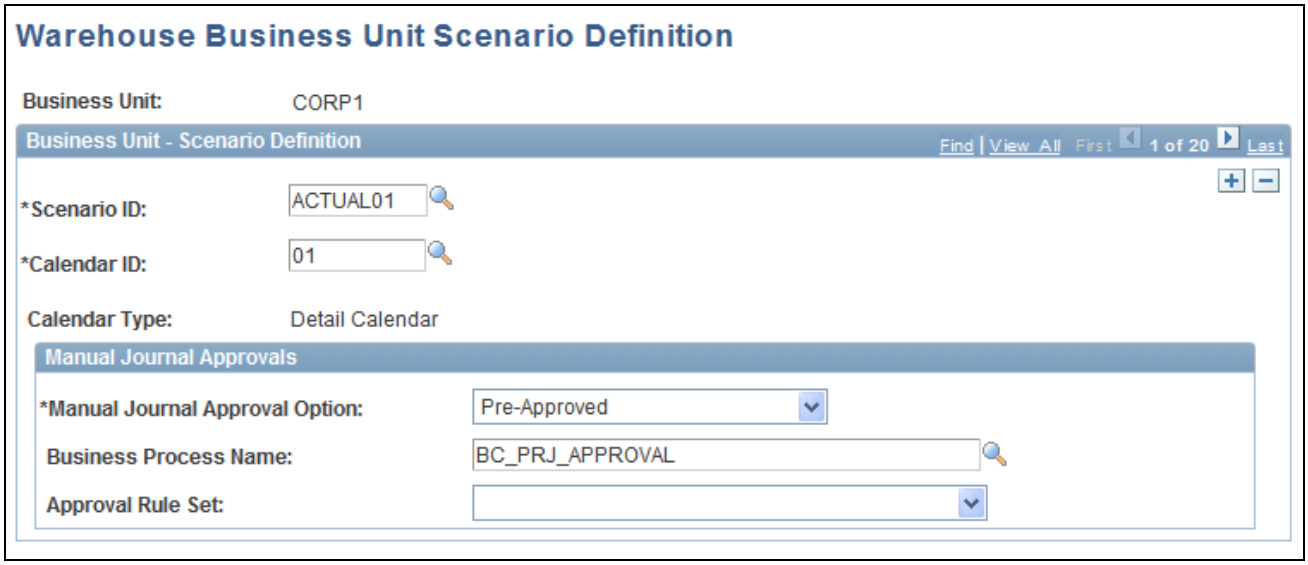
Entering Forecast Scenario Economic Assumptions

Access the Scenarios - Economic Assumptions page (EPM Foundation, Business Metadata, Business Framework, Scenarios, Economic Assumptions).

The fields on this page are the same as those discussed for the Historical Forecast - Economic Assumptions page.

Assigning Scenarios to Warehouse Business Units

Access the Warehouse Business Unit Scenario Definition page (EPM Foundation, Business Metadata, Business Framework, WBU Scenario Definition, Warehouse Business Unit Scenario Definition).



Warehouse Business Unit Scenario Definition page

Scenario ID For each business unit, select the appropriate scenario ID.

Calendar ID For each business unit, select the appropriate calendar ID.

Note. There is a relationship between the calendar ID on this page and the calendar ID on the Scenario Definition page. If you make an entry on this page that associates a calendar ID to a scenario and business unit combination, you can no longer change the calendar ID on the Scenario Definition page (the field is disabled).

Manual Journal Approval Journal approvals allow specific requirements to be met before the approval is finalized. The system defaults to *Pre-Approved* for the Manual Journal Approval option. The *Approval Required - Security* and *Approval Required - Workflow* options are used by PeopleSoft Enterprise Global Consolidations.

Business Process Name If you are using the approval workflow option in PeopleSoft Enterprise Global Consolidations, specify the business process name for the approval workflow.

Approval Rule Set If you are using the approval workflow option in the PeopleSoft Enterprise Global Consolidations analytical application, specify the approval rule set to be used for the selected business process.

See *PeopleSoft Enterprise Global Consolidations 9.1 PeopleBook.*

Chapter 20

Streamlining Processing with Jobstreams

This chapter provides an overview of jobstreams and discusses how to:

- Set up chunking.
- Work with engine metadata.
- Set up job metadata.
- Set up jobstreams.
- Work with record suites.
- Create additional instances of temporary tables.
- Remove extraneous temporary tables from record suites.
- Run jobstreams.
- Track jobs.
- View engine messages.

Understanding Jobstreams

This section discusses:

- Jobstreams.
- Jobstream terminology
- Jobstream processing.
- Jobstream record suites.
- Jobstream chunking.
- Spawn process.
- Resolver engine.
- Resolver and chunking.
- Process monitor.

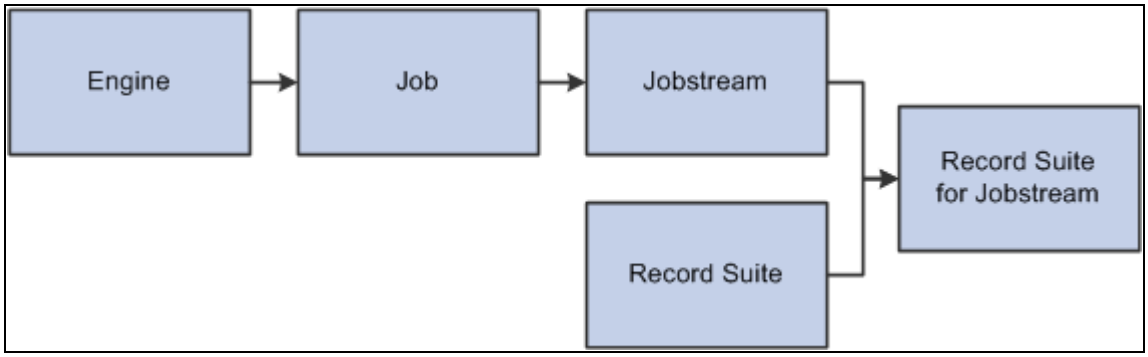
Jobstreams

To help streamline your processing, PeopleSoft provide jobstreams which use temporary tables for intermediate processing. Jobstreams enable different users to run their own jobs using instances of the same processing engines at the same time. Jobstreams enhance performance by sharing temporary tables passed between jobs.

Instead of locking up the fact (primary input) tables, jobstreams use temporary tables for intermediate processing. A set of delivered temporary tables, referred to as a record suite, is assigned when the first job of a jobstream is run, and then the tables are released when the last job of a jobstream is completed. The use of record suites frees up the fact tables so that another user can access them and run a concurrent job. Each job then has its own record suite for a jobstream.

There are several steps involved in setting up a jobstream. PeopleSoft delivers predefined processing engines and engine metadata, jobs and job metadata, jobstreams, and record suites. If you use the predefined metadata, the only item that you have to specify before you run an engine is one or more record suites for each jobstream for a given setID.

The following diagram illustrates the components that make up a jobstream:



Jobstream overview

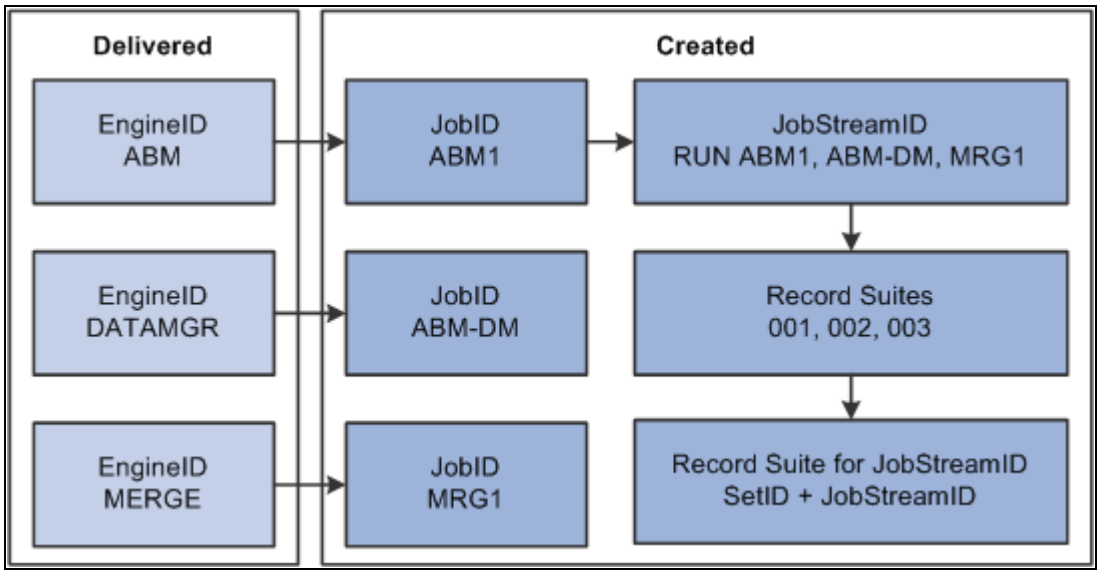
Jobstreams work by creating a copy of the processing engine. When you run a jobstream, you can:

- Run multiple engines sequentially in one jobstream.
- Run each individual engine in its own jobstream.
- Run one sequential jobstream for multiple fiscal years or accounting periods.

Suppose you want to run the Activity-Based Management engine, Data Manager engine, and Merge engine at the end of the business day. You can select a jobstream that runs all three engines automatically. Each engine runs sequentially and populates specific temporary tables, with the Merge engine transferring the data from the temporary tables back to the fact tables. You only need to initiate the jobstream, no further action is required.

The Merge (PF_MERGE) engine merges the output temporary tables into the final tables for use as input for other processes. PF_MERGE is the last job in all jobstreams except when the POST job is run at the end of a jobstream.

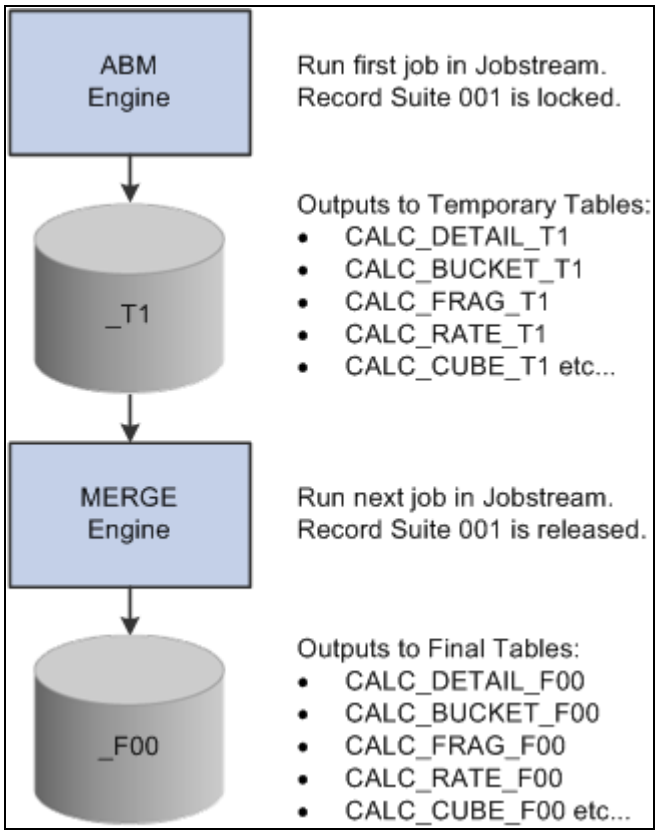
The following diagram illustrates how jobstream setup works:



Jobstream setup

In the above diagram, note that the engine IDs on the left side of the illustration are delivered with PeopleSoft EPM. You create the job and jobstream IDs, and then assign record suites to the jobstream.

The following diagram illustrates how the Merge engine moves output from the Activity-Based Management (ABM) engine to the into final fact tables:



Merge engine process

Jobstream Terminology

The following terms apply to jobstreams in PeopleSoft EPM applications:

Engine Metadata Identifies the PeopleSoft application engine process that you want to run by engine ID.

Job Metadata Enables you to create an instance or copy of a PeopleSoft application engine program to use in your jobstream.

Jobstream Enables you to combine job IDs into a jobstream ID to pass data from one job to the next. Every job you run must be in a jobstream, whether it is a combination of sequential jobs or a single job. All jobs in the jobstream must be run for the same parameters (business unit, scenario, fiscal year, and so on).

Record Suites Consist of a group of temporary tables with the same temp table append for processing instances of an application engine. Temporary tables are used during engine processing to free up the original fact tables so other users can access them and provide faster run times. A jobstream has exclusive use of the record suite during its execution, so there is no table locking or sharing.

For example, you might create a jobstream to first run the ABM engine, then the Data Manager engine, and finally run the Merge process. Each engine runs sequentially and populates specific temporary tables, with the Merge engine transferring the data from the temporary tables back to the fact tables.

Jobstream Record Suites Assign record suites to the jobstream ID you create (using the appropriate setID). The setID used must be the same setID assigned to the run control parameter business unit under record group PF_03.

Jobstream Processing

When you run a jobstream, the following occurs:

- Record suite 001 is locked and the first engine runs placing its output into the appropriate temporary table in record suite 001.
- The next engine runs getting its input from the temporary tables generated by the first job and putting its output into other temporary tables in record suite 001.
- The last job in the jobstream (either PF_MERGE or PF_POST) reads the data in the temporary tables, merges it, and then writes the output to the final target tables. The system releases the record suite.

Note. Not all processes use jobstreams. For example, PF_SUMM does, but PF_MODIFICATION does not. Exceptions are noted in the PeopleBooks documentation as appropriate.

Jobstream Record Suites

When you initially process a jobstream, the system checks which record suites are assigned to the specified setID and jobstream ID.

This process enables you to reserve a set of record suites for a specific type of engine. For example, if record suites 001 and 002 are assigned to PeopleSoft Enterprise Activity-Based Management (ABM) jobstreams, and 003 is authorized for PeopleSoft Enterprise Asset Liability Management (ALM), then PeopleSoft ALM jobstreams never compete for record suite availability with ABM jobstreams.

Jobstream Chunking

Chunking is a mechanism that enables you to select a smaller chunk of data for further processing and to parallel process data in multiple chunks. It enables you to horizontally partition source data so that only a subset of data is processed by an engine. The enables users to run multiple engines with different criteria and to run them in parallel to reduce the processing time.

During a jobstream run, chunking occurs when technical scenario is associated with the run scenario based on the scenario selected on the Technical Scenarios page. After a jobstream identifies that chunking has been requested, the jobstream initiates the PF_CHUNK application engine program to process each chunking selection. The jobstream then invokes a parallel application engine PF_SPAWN to process each chunking selection. This program spawns a job for each chunking definition.

The number of jobs that can be spawned in parallel is restricted to the number of available record suites. You require one record suite for the jobstream process and one for each of the spawned processes.

Because all jobs that use the same technical scenario may not require chunking, the decision to chunk is based on the chunking selection in the engine metadata and chunking criteria specified on the Technical Scenarios page.

Spawn Process

The *spawn* application engine process (PF_SPAWN) provides greater control over jobstream processing by enabling jobs to be launched as needed.

Spawn Process Tables

PF_SPAWN creates the following tables to store data while the jobstream runs:

- The PF_SPWN_JOB_TBL table stores all required information about spawned jobs.

  Entries in this table are deleted once all spawned jobs are complete.

- The PF_SPWN_CTL_T temporary table passes the run control parameters for each spawned job.

  Use the sequence number field to control the order of the spawned jobs.

- The PF_SPWN_CRIT_T table stores all of the required spawn criteria for each of the spawned jobs.

Resolver Engine

The Resolver engine further enhances and increases application performance by reducing the amount of data an engine needs to process. You do not invoke the Resolver (PF_ENG_PROC.RESOLVE) engine. The system invokes it automatically as part of startup processing to resolve all records and tablemaps specified in the engine metadata for an engine. An application needs to explicitly invoke the Resolver engine to resolve datamaps, filters, constraints, and data sets referenced in business rules.

Resolution occurs on the setID, business unit, scenario ID, effective date, as of date, fiscal year, and accounting period. The resolution process only moves the data that matches the run control values from the table to the associated temporary table as defined in the record metadata. The engine works only on the data in the temporary table.

Individual engines call the Resolver engine as part of their run processes.

Note. The Resolver engine is limited to resolving tables for only one value at a time. For example, it cannot resolve for multiple business units.

Resolver and Chunking

As part of chunking, the Resolver engine applies chunk criteria to the record that is being chunked based on the criteria defined on the Technical Scenarios page. When the system invokes the Resolver engine, it checks all the records that it needs to resolve to see if the record requires chunking. If this is the case, the Resolver engine checks the record to see if the chunk field exists in the record. If the field exists, the system appends chunk criteria to the resolver query for this record.

Below is an example of chunk criteria:

```
(CUST_ID IN (SELECT CUST_ID FROM PS_CUSTiINTFC_F00 WHERE CUST_ID BETWEEN ⇒
('1000','10000'))
```

In this case the CUST_ID is the chunk field that exists in the record being resolved.

Process Monitor

During job processing, use Process Monitor to review the status of reports and processes. You can monitor process requests, server status, and the status of any job in the queue. If there are messages related to a process, you can view them from Process Monitor, as well. For example, if a process encounters an error, or if a server is down, you can find out almost immediately.

See *PeopleSoft Enterprise PeopleTools PeopleBook: PeopleSoft Process Scheduler*

Failed Jobstreams and the Process Monitor

When a jobstream fails, one of the following status messages appear:

- *Error*: Indicates that the program that is associated with the process request encountered an error while processing transactions within the program. In this case, delivered programs are coded to update the run status to Error before terminating.

- *No Success*: Indicates that the program encountered an error within the transaction. No Success is different from Error because the process is marked as restartable.
- *Success With Application Error*: Indicates that a jobstream has completed, but with an application error. For example, a jobstream may result in an application error due to unavailable record suites.

Setting Up Chunking

This section discusses how to:

- Establish chunking in the engine metadata.
- Set up chunking criteria.

Establishing Chunking in the Engine Metadata

To set up chunking you access the Engine Metadata - State Variables or Engine Metadata - Source TableMaps pages to set up application engine parameters. On the State Variables page, you specify the records to be chunked during the resolve process. Alternatively, on the Source TableMaps page you can specify the tablemaps to be chunked. The next section in this chapter details the pages in the Engine Metadata component on which you define this setup.

As the next step, go to the Technical Scenarios page to set up the chunking criteria. Technical scenarios enable you to set up the object type values that the Resolver uses to chunk the records and tablemaps you are resolving. You establish the records or tablemaps to resolve on the Engine Metadata - State Variables and Engine Metadata - Source TableMaps pages.

See Also

Chapter 20, "Streamlining Processing with Jobstreams," Working with Engine Metadata, page 469

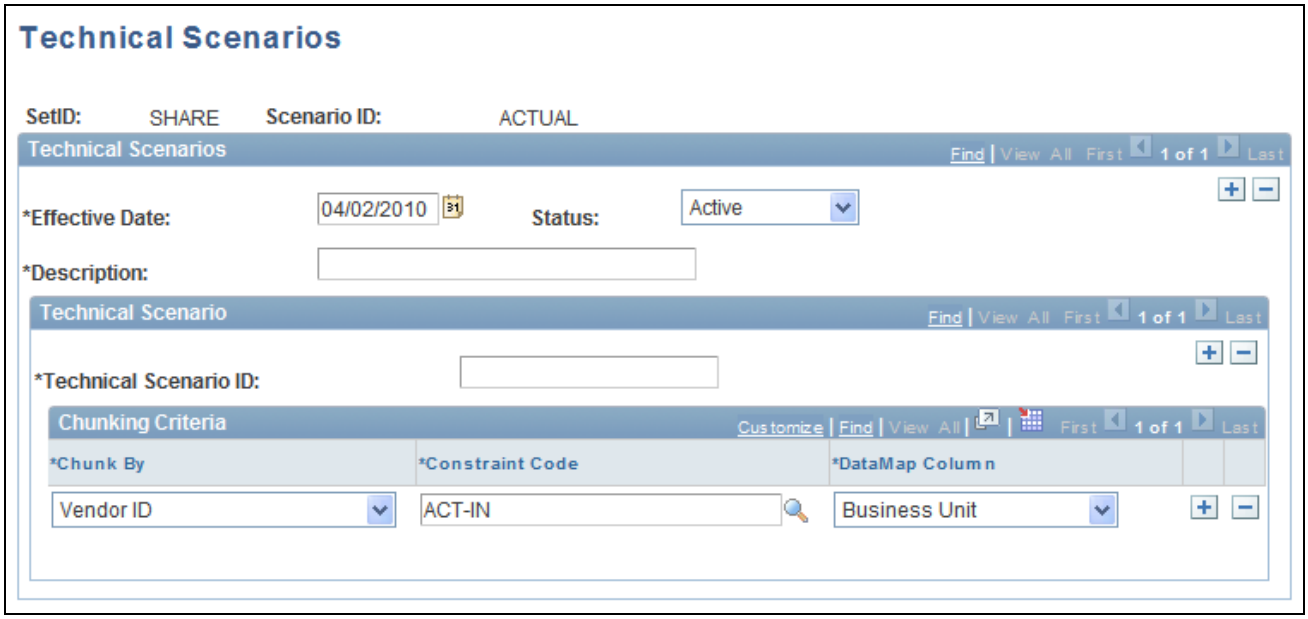
Pages Used to Set Up Chunking

Page Name	*Definition Name*	*Navigation*	*Usage*
Engine Metadata - State Variables	PF_META_ENG_TBL2	EPM Foundation, Job Processing, Setup Engines and Jobstreams, Engine Metadata, State Variables	Specify records to be chunked during the resolve process.
Engine Metadata - Source TableMaps	PF_META_ENG_TBL4	EPM Foundation, Job Processing, Setup Engines and Jobstreams, Engine Metadata, Source TableMaps	Specify source tablemaps to be chunked during the resolve process.

Page Name	*Definition Name*	*Navigation*	*Usage*
Technical Scenario	PF_CHUNK_DFN	EPM Foundation, Business Metadata, Business Framework, Technical Scenarios	Set up the object type values that the Resolver uses to chunk the record or tablemap you are resolving.

Setting Up Chunking Criteria

Access the Technical Scenarios page (EPM Foundation, Business Metadata, Business Framework, Technical Scenarios).



Technical Scenarios page

Scenario ID Select the scenario ID to which to link this technical scenario. When a jobstream runs for this scenario, the chunking is invoked.

Technical Scenario ID Enter the technical scenario ID to identify the chunking selection. You may enter as many chunk codes as you like for each scenario. Each chunk code may have different chunk criteria.

Chunk By Select the object type to use for chunking. You must select the object type you selected in the Engine Metadata component.

Note. You can apply only one chunking criteria to a tablemap. The system does not support duplicate object types.

Constraint Code Select the constraint that filters the values for your objects at run time.

DataMap Column Select one column from the datamap of the constraint. This should be, but is not restricted to, the column that matches the object type you have selected. For example, if your chunk object type is *PRODUCT ID,* then your datamap column from the constraint may be *PRODUCT_ID* or *PRODUCT_TYPE.* You may copy chunking criteria to another effective date for the same scenario only. You need to be very careful about defining chunk criteria. You need to make sure to define your chunks to cover a complete set of data without any duplicates.

Note. If you delete a scenario with chunking criteria, the chunking criteria are automatically deleted.

You can enter multiple technical scenarios to process data in multiple chunks. Remember, that all technical scenarios should select a mutually exclusive data set. The data is also not necessarily processed in the order defined.

Warning! If you enter an invalid or duplicate constraint code, the jobstream abends at run time.

You have established the chunk objects and the chunking criteria. Chunking initiates when there is a technical scenario associated with the run scenario.

For chunking to be successful, you must ensure that:

- The object type selected in the Engine Metadata component matches the one in the technical scenario used by the engine. This initiates the chunking process.
- The column you select for the chunking criteria on the Technical Scenarios page must be in the record or any record of a tablemap you have selected for chunking. This completes the chunking process.

Note. You can only resolve (chunk) a record once in an engine. The system resolves tablemaps first. Any records resolved as part of the tablemap are not resolved again in the state record.

Working with Engine Metadata

PeopleSoft EPM delivers predefined engine metadata. Unless you create your own application engine processes as part of your implementation, you do not need to create any engine metadata. However, you do need to create different instances of the delivered engines to enable parallel processing, described in the next section.

In most cases, you only use the pages described in this section to review and modify the delivered engine metadata. You may also use them to define chunking selections.

If you do add an application engine process, use the pages described to add the engine metadata for the new application engine.

This section discusses how to:

- Review, modify, or add engine metadata.
- Review, modify, or add state variables.
- Define rule state variables.

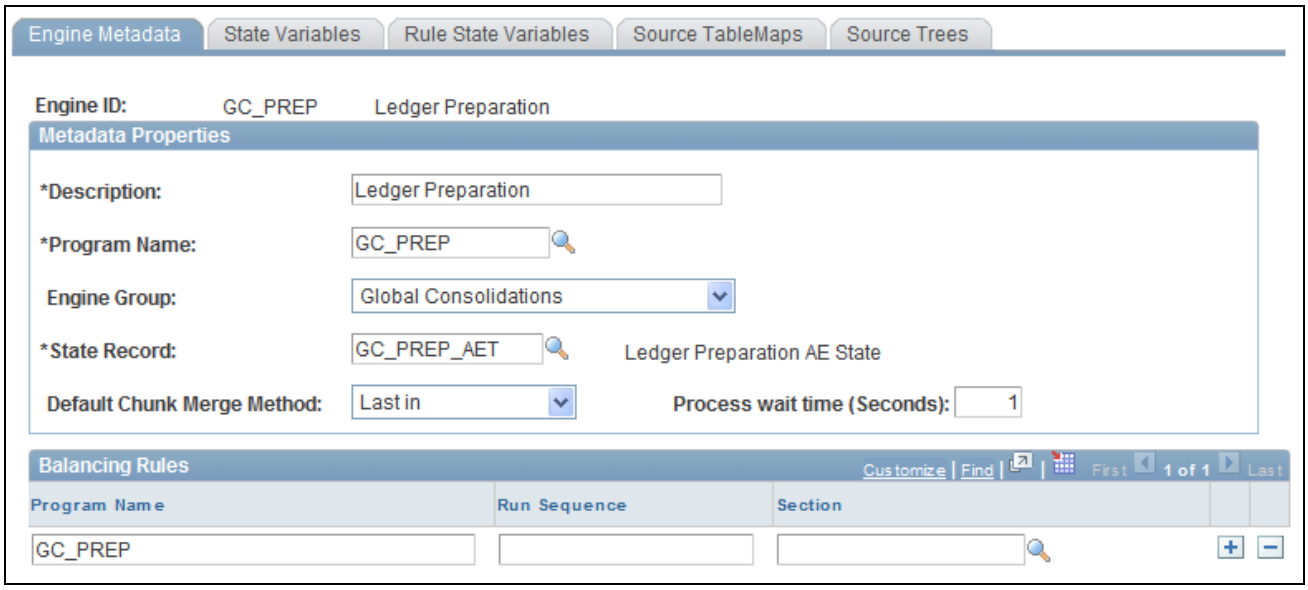
- Specify source tablemaps.
- Specify source trees.

Pages Used to Work with Engine Metadata

Page Name	*Definition Name*	*Navigation*	*Usage*
Engine Metadata	PF_META_ENG_TBL1	EPM Foundation, Job Processing, Setup Engines and Jobstreams, Engine Metadata	Review delivered engine metadata and modify if required. Add new metadata for a new application engine process. Unless you create your own application engine process, you do not need to create any engine metadata.
Engine Metadata - State Variables	PF_META_ENG_TBL2	EPM Foundation, Job Processing, Setup Engines and Jobstreams, State Variables	Review state variables modify if required. These state variables enable dynamic changes to application engine inputs.
Engine Metadata - Rule State Variables	PF_META_ENG_TBL3	EPM Foundation, Job Processing, Setup Engines and Jobstreams, Rule State Variables	Define rule state variables. This enables dynamic changes in the application engine.
Engine Metadata - Source TableMaps	PF_META_ENG_TBL4	EPM Foundation, Job Processing, Setup Engines and Jobstreams, Source TableMaps	Specify source tablemaps if the engine needs to resolve a set of tables before running.
Engine Metadata - Source Trees	PF_META_ENG_TBL5	EPM Foundation, Job Processing, Setup Engines and Jobstreams, Source Trees	Specify source trees to be flattened during the application engine initialization phase.

Reviewing, Modifying, or Adding Engine Metadata

Access the Engine Metadata page (EPM Foundation, Job Processing, Setup Engines and Jobstreams, Engine Metadata).



Engine Metadata page

Engine ID Displays the unique identifier for the application engine.

Program Name Select a delivered application engine program.

Engine Group Select an engine group to tie the engine to a column on the Scenario definition page. When an engine is run, it picks up the model ID from the Scenario Definition page and uses it to process the rules and data.

State Record Displays the name of the state record. The state record defines which record is used to define state variables on the next page in this component. It is a prompt for the state variable column on that grid.

Default Chunk Merge Method Select a value to determine the method for the reloads of a chunked job merge to the main jobstream.

Values are:

Last in: Merges any duplicate data last, replacing the old data.

First in: Merges any duplicate data so that the original data remains.

Aggregate: Merges any duplicate data and aggregates it.

Append: Appends any duplicate data.

Process Wait Time (Seconds) Displays the seconds of lag time before the next process runs. The default for this field is set on the Installation Options - Web Services page. You can override the default setting here if required.

See Chapter 4, "Setting Up EPM Business Rules," Setting Web Services Options, page 49.

Balancing Rules Enter any balancing rules that you want to run with this engine.

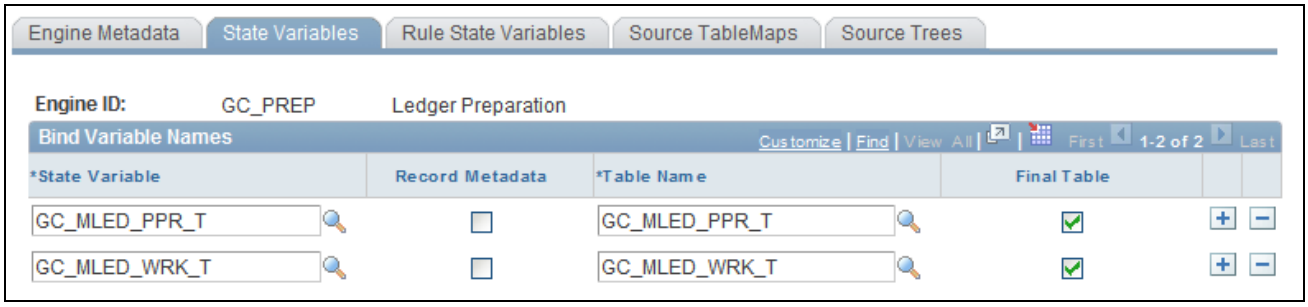
The program name appears.

Run Sequence Displays the sequence number for the run. This number must be unique.

Section This is not a required field. It is used to help you focus on the problem area when there is an out-of-balance situation. Before a section can be considered valid on the job totals page, it must be defined in the engine metadata. Valid section codes to be entered on the Engine Metadata page are the actual application engine section within the application engine program.

Reviewing or Modifying State Variables

Access the Engine Metadata - State Variables page (EPM Foundation, Job Processing, Setup Engines and Jobstreams, State Variables).



State Variables page

PeopleSoft delivers predefined state variables that enable dynamic changes to application engine inputs. State variables can include the input and output tables that an engine requires, so that the table name does not have to be hard-coded in the application engine program. You rarely need to change state variables. However, you can select records for chunking.

In the Bind Variables Names group box specify the following:

State Variable Specify the record field name. The prompt list only shows those state record fields for the state record specified on the Engine Metadata page.

When the engine runs, the record stub is stored in this field on the state record. The page must be saved before the prompt works.

Record Metadata Select this option if this is record metadata.

Table Name Specify the table that is used in the application engine to populate the state variables with the record stub.

Final Table Select this option if this is the final output table.

Chunk Select this option to enable chunking for this record.

The Chunk By and Merge Method fields display.

Chunk By Select the object type for the chunking for example by *Group ID*, *Job Code*, *Model ID*, and so on. This field defaults to *Group ID.*The object type you select here must match that selected on the Technical Scenarios page.

Merge Method Select a value to determine the method for the reloads of a chunked job merge to the main jobstream.

Values are:

Last in: Merges any duplicate data last, replacing the old data.

First in: Merges any duplicate data so that the original data remains.

Aggregate: Merges any duplicate data and aggregates it.

Append: Appends any duplicate data.

You set the default on the Engine Metadata page.

Warning! If at least one chunk check box is selected, the Merge Method list box appears for the tables that are marked as *Final*. You may override the default merge method.

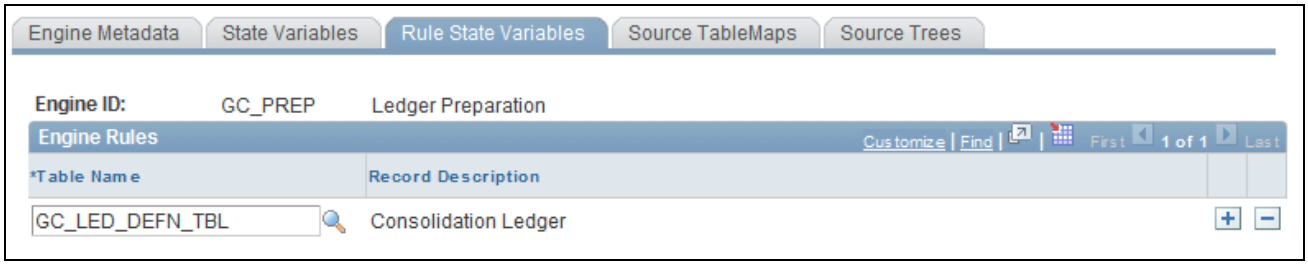
Record Stub and Table Status

The following table defines the relationship between record metadata and table status:

Record Metadata Selected?	*Final Table Selected?*	*Chunk Selected?*	*Information*
Yes	No	Yes	Typical input. Record stub is put into state record and table is resolved.
Yes	Yes	Yes	Typical output. Record stub is put into state record, temporary table is truncated, and the table is marked so that it is not resolved in the jobstream.
No	Yes	No	Record stub is put into state record and temporary table is truncated.
No	No	No	Record stub is put into state record.

Defining Rule State Variables

Access the Engine Metadata - Rule State Variables page (EPM Foundation, Job Processing, Setup Engines and Jobstreams, Rule State Variables).



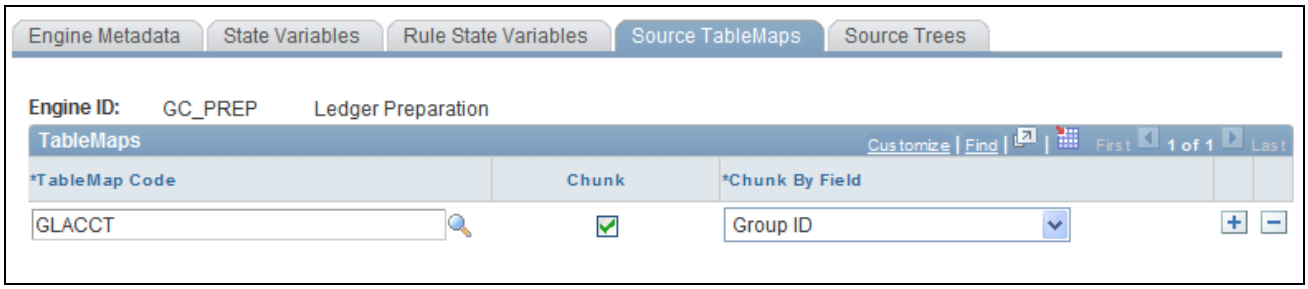
Rule State Variables page

Table Name Select the parent table of the rule to be resolved. You identify only the parent table because both parent and child tables are resolved. This rule table is resolved automatically as part of running the application engine.

The description for the table appears.

Specifying Source TableMaps

Access the Engine Metadata - Source TableMaps page (EPM Foundation, Job Processing, Setup Engines and Jobstreams, Source TableMaps).



Engine Metadata - Source TableMaps page

TableMap Code Displays only predefined tablemaps that are available for selection. The selected tablemap is automatically resolved as part of the application engine execution.

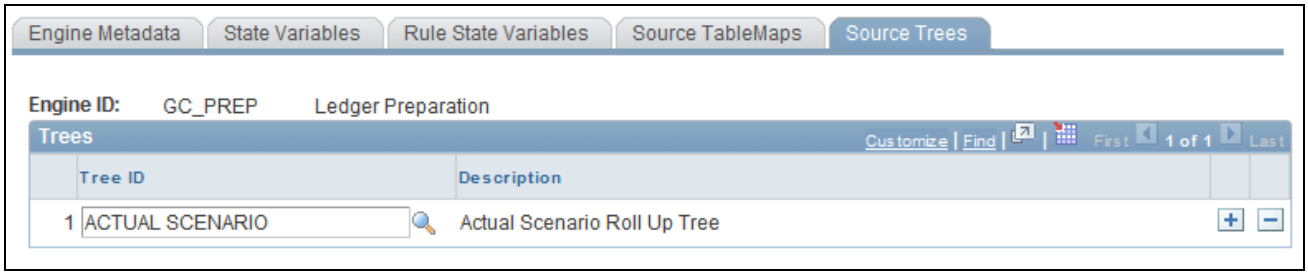
Chunk If you select this check box, you enable the chunking of a tablemap within the Resolver. The Chunk By Field automatically displays a default value of *GROUP_ID*

Chunk By Field Select an object type for chunking for example by *Group ID, Job Code, Model ID,*and so on. This field defaults to *Group ID.*

This object type must match the type selected on the Technical Scenarios page.

Specifying Source Trees

Access the Engine Metadata - Source Trees page (EPM Foundation, Job Processing, Setup Engines and Jobstreams, Source Trees).



Source Trees page

Tree ID Select the trees to be resolved during the application engine initialization phase. If the run control parameters do not match the tree parameters, the application engine logs a message that the tree is not resolved.

Setting Up Job Metadata

Job metadata (in other words, job ID) represents an instance, or copy, of a PeopleSoft application engine program and is used in the creation of a Jobstream. A job ID can be reused multiple times in the same jobstream or across multiple jobstreams. This enables you to use the same application engine more than once without having to define multiple job IDs. PeopleSoft deliver predefined job IDs for many of the jobs you need to run, but you can create additional job IDs if necessary.

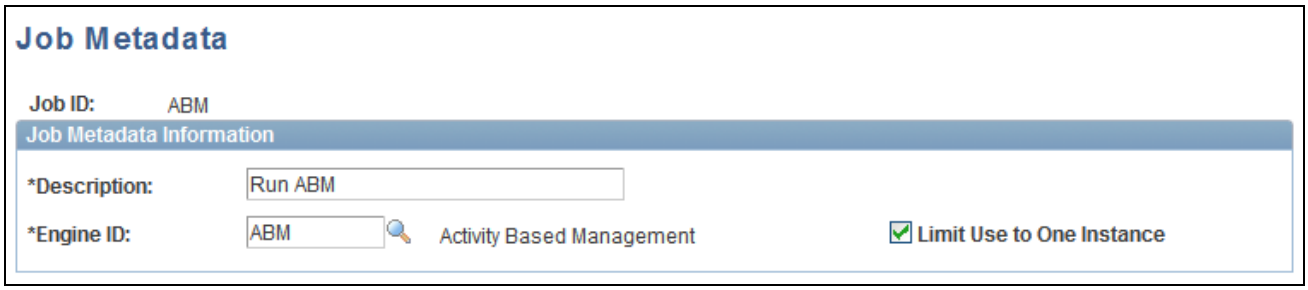
After you complete the creation of a job ID, you can create a jobstream that runs just one engine or a combination of engines sequentially.

Page Used to Set Up Job Metadata

Page Name	*Definition Name*	*Navigation*	*Usage*
Job Metadata	PF_META_JOB_TBL1	EPM Foundation, Job Processing, Setup Engines and Jobstreams, Job Metadata	Create a job ID to define an instance an engine.

Creating a Job ID

Access the Job Metadata page (EPM Foundation, Job Processing, Setup Engines and Jobstreams, Job Metadata).



Job Metadata page

Job ID Displays the unique identifier for the instance of the engine.

Engine ID Select the correct engine ID for the instance you are defining.

Limit Use to One Instance Select this check box to make a job ID unique.

If this check box is selected, the job ID specified cannot be reused. You should deselect this check box for Job IDs that you create.

Setting Up Jobstreams

All jobs must be placed in a jobstream—you can add a single job or combine multiple jobs and pass data from one job to the next. All engines in a jobstream must run with the same run parameters (business unit, scenario ID, fiscal year, and so on). Additionally, jobstreams use the table appends defined on the Record Suites page to enable parallel processing of the same engines and tables by multiple users.

PeopleSoft EPM is delivered with a number of predefined jobstreams. Refer to your application-specific PeopleBooks for details on these jobstreams.

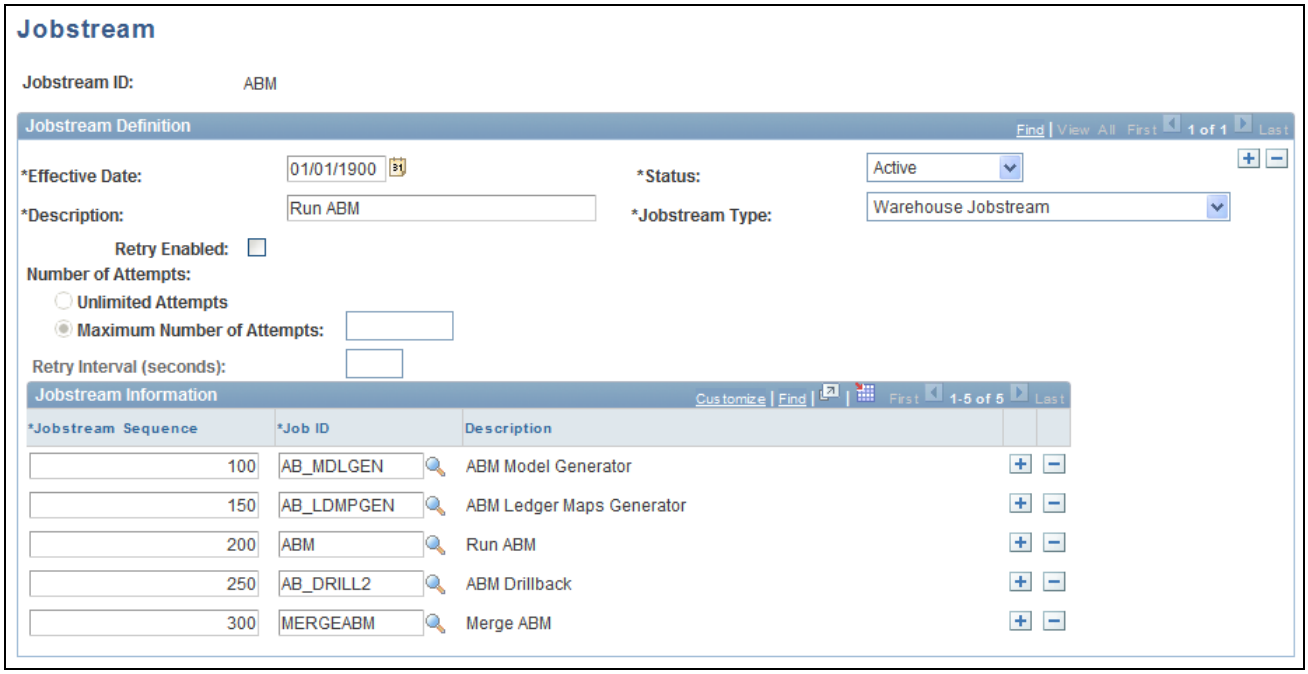
If a job in a jobstream cannot access a locked record suite during processing, the jobstream is sent to queue and waits for reprocessing once the record suite becomes available. You specify the queuing method on the Jobstream page.

Page Used to Define a Jobstream

Page Name	*Definition Name*	*Navigation*	*Usage*
Jobstream	PF_JOBSTRM_DFN1	EPM Foundation, Job Processing, Setup Engines and Jobstreams, Processes in Jobstream, Jobstream	Define a jobstream by specifying which engines (job IDs) to run and the order in which to run them.

Defining Jobstreams

Access the Jobstream page (EPM Foundation, Job Processing, Setup Engines and Jobstreams, Processes in Jobstream, Jobstream).



Jobstream page

Jobstream ID Displays the unique identifier for the series of jobs to be run.

Jobstream Type Select the type of jobstream you are defining.

Retry Enabled? Select this check box if you want a jobstream to queue if its related record suite is locked.

If this check box is selected the jobstream queues automatically when its record suite is locked and reruns when the record suite becomes available.

Number of Attempts

Unlimited Attempts Select this option to have a queued jobstream attempt reprocessing indefinitely.

Maximum Number of Attempts Select this option if you want a queued jobstream to attempt reprocessing a limited number of times.

In addition, specify the number of times you wish to have a queued jobstream attempt reprocessing. Once this option is selected, the Retry Interval (seconds) field becomes available for input.

Note. After the queued jobstream reaches the maximum number of reprocessing attempts specified in this field, the jobstream fails and a Warning status is set in the Process Monitor.

Retry Interval (seconds) Specify the amount of time (in seconds) that should elapse between each reprocessing attempt.

Jobstream Information

Jobstream Sequence Enter a unique number, such as *100,* for the first job ID this jobstream should run. The next job ID to run is *200.*

Note. The actual sequence number is not important; it represents the sequence in which you want to run jobs. The sequence number must be unique.

Job ID Select the job ID form the drop-down list box. The Job ID is created on the Job Metadata page and represents a unique instance of an engine for this jobstream.

After you create a jobstream, you can go back to the Job Metadata page and verify the Jobstream ID and Job Use fields. They are now populated.

Linking Jobstreams Sequentially

You can link multiple jobstreams sequentially using PeopleTools *JobSet* functionality. JobSets enable you to schedule any application engine process using a schedule JobSet definition. Because a jobstream is an application engine process, you can use the jobset to sequentially link multiple jobstreams together. Using the JobSet functionality you can:

- Use different run control IDs for each process within a jobset.
- Run processes from different operating systems or servers.
- Monitor processes from the process scheduler.

See *PeopleSoft Enterprise PeopleTools PeopleBook: PeopleSoft Process Scheduler*

Working with Record Suites

PeopleSoft delivers three predefined record suites: 001, 002, and 003. Record suites are a group of temporary tables with the same temp table append for processing instances of an application engine. In order for your jobstream to run, you must associate record suites with the jobstream ID. You can also create new record suites and add them.

This section discusses how to:

- Create new record suites.
- Set up record suites.
- Associate record suites with a jobstream.

Pages Used to Work with Record Suites

Page Name	*Definition Name*	*Navigation*	*Usage*
Record Suite	PF_RECSUITE_DFN1	EPM Foundation, Job Processing, Setup Record Suites, Define Record Suite, Record Suite	Set up the record suites that are delivered with PeopleSoft Enterprise Performance Management. You might need to create more temporary tables.
Jobstream Record Suites	PF_REC_JOB_TBL1	EPM Foundation, Job Processing, Setup Record Suites, Jobstream Record Suites, Jobstream Record Suites	Associate record suites with a jobstream. You usually assign all three record suites to each jobstream ID.

Creating Record Suites

You can create record suites in addition to those delivered with PeopleSoft EPM.

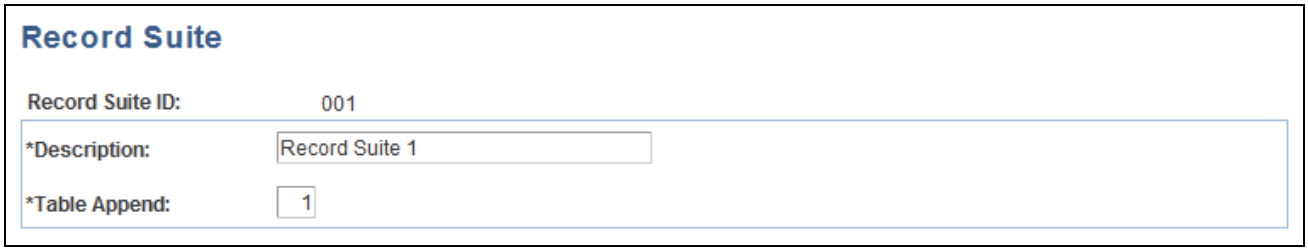
To create new record suites:

1. Select PeopleTools, Utilities, Administration, PeopleTools Options.
2. Increase the number of temp table instances to the desired number.
3. Add new record suites on the Record Suite page described below.
4. Build the EPM_TEMP_RECORDS project.

Note. Be aware that this also builds 001, 002, and 003 again.

Setting Up Record Suites

Access the Record Suite page (EPM Foundation, Job Processing, Setup Record Suites, Define Record Suite, Record Suite).



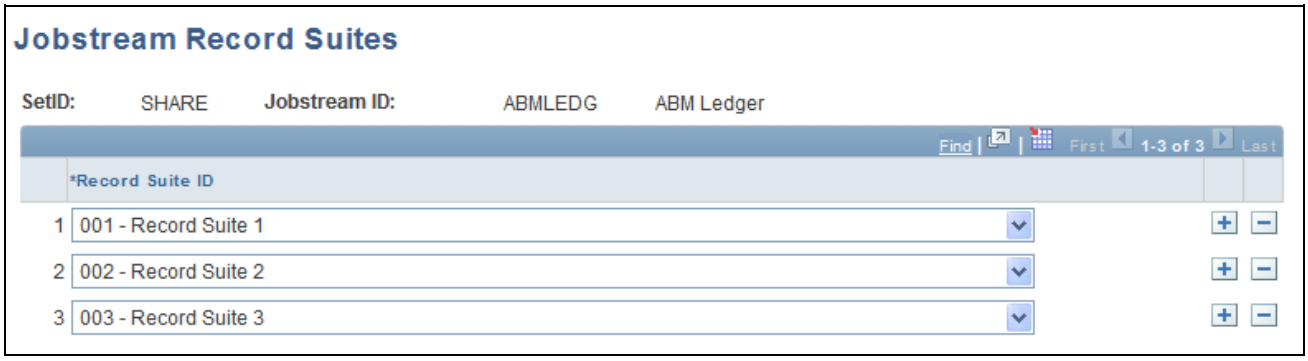
Record Suite page

Record Suite ID Displays the unique identifier for the record suite.

Table Append Enter the number to be used as the table append for the record suite. The number must be less than or equal to the number that is defined in the temp table instances field on the PeopleTools, Utilities, Administration PeopleTools Options page. You may define chunking record suites as well.

Associate Record Suites with a Jobstream

Access the Jobstream Record Suites page (EPM Foundation, Job Processing, Setup Record Suites, Jobstream Record Suites, Jobstream Record Suites).



Jobstream Record Suites page

SetID Displays the setID that you pointed to for your record group.

For example, if the run control is for business unit CORP1 and it points to setID MODEL, you must create a jobstream and record suite combination for MODEL.

Jobstream ID Displays the jobstream ID to which you are assigning the record suites.

Record Suite ID Select the record suite to be used by the jobstream. You can add more than one record suite.

Creating Additional Instances of Temporary Tables for Record Suites

You can add instances of temporary tables to a record suite per your business requirements. To create additional instances of the temporary tables:

1. In Application Designer copy all tables in the relevant project and change the last two characters of the table name.
2. Rebuild the project.

   Your new temporary table suite is ready. Repeat this process to create additional temporary table suites.

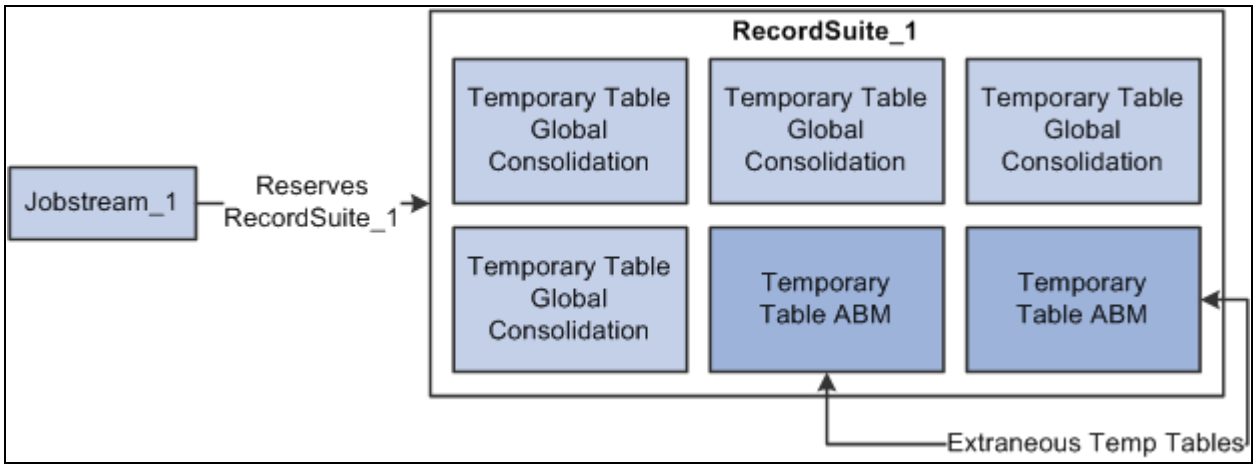
Project Name	*Description*
EPM_SECURITY	All delivered views that give secure access to EPM objects.
EPM_TEMP_RECORDS	One instance of the temporary tables needed to run all the Enterprise Performance Management engines.
EPM_TEMP_RECORDS_INC	Incremental project that contains one instance of newly added temporary tables only.
EPM_TOOLS	All EPM-specific changes to the PPLTOOLS project.

Note. If you are altering the number of temporary table instances, change the default setting of *3* to the desired number on the PeopleTools Option page (located at PeopleTools, Utilities, PeopleTools Option) and rebuild EPM_TEMP_RECORDS project in PeopleSoft Application Designer.

See *PeopleSoft Enterprise PeopleTools PeopleBook: PeopleSoft Application Designer Developers Guide*

Removing Extraneous Temporary Tables from Record Suites

Jobstreams use record suites and their corresponding temporary tables to process data. PeopleSoft delivers record suites with an entire set of EPM temporary tables, regardless of the products you license. For example, if you only purchase the Global Consolidations analytical application, the delivered record suites still contain temporary tables for all other EPM products—such as ABM and Budgeting. The following diagram depicts this scenario:

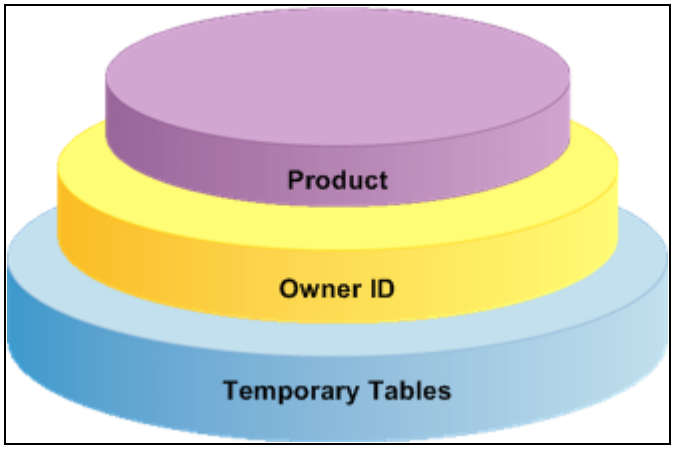


Record suite with extraneous temporary tables

Each delivered record suite can potentially contain hundreds or thousands of extraneous temporary tables—due to the fact that the total number of EPM temporary tables exceed 3,300. Each time a record suite is processed with these superfluous temporary tables, processing efficiency is severely degraded. However, PeopleSoft provide functionality that enables you to remove unnecessary temporary tables from the EPM database

Understanding the Temporary Table Removal Process

Extra temporary tables are removed from the EPM database by running the Clean Temporary Tables application engine (PF_CLEAN_TMP). To identify the temporary tables that need to be removed from EPM, all temporary tables are assigned an ownerID. Each owner ID is assigned to a specific EPM product. While temporary tables can only be assigned to one owner ID, you can assign several owner IDs to a single product. The following diagram depicts the hierarchical relationship between temporary tables and the products.



Temporary table to product hierarchy

The Clean Temporary Tables process uses the product, ownerID, and customer licensing information (plus the relationships between these objects), to identify the tables that should be removed from the EPM database. The information is stored in the following EPM tables:

- PS_PF_PROD_TO_OWNR: Contains product code, ownerID, and record type, and identifies the relationship between EPM product and ownerID.
- PSRECDEFN: Identifies the relationship between EPM temporary tables and ownerID.
- PSINSTALLATION: Identifies which EPM products you have licensed.

At runtime the Clean Temporary Tables application engine uses the information stored in the aforementioned tables to identify and delete all instances of temporary tables that are not required by your licensed product(s). It then regenerates the EPM_TEMP_RECORDS and EPM_TEMP_RECORDS_INC application designer projects with the new temporary tables.

All temporary tables being shared among applications belong to a special ownerID named *AppCommon* and are not deleted. There are also fundamental temporary tables associated with the ownerID *EPM Foundation* and are not deleted.

After running the Clean Temporary Tables process, all temporary tables that have been dropped from the database are logged in the PF_TMPTBL_LOG table. You can view the results of this process using the Cleanup Log page.

The Clean Temporary Tables application engine should be run after every install, upgrade or patch of PeopleSoft EPM products. After it runs, open the EPM_TEMP_RECORDS_INC project in Application Designer, and if not empty, rebuild the project to ensure that newly added temporary tables are built.

Delivered EPM_TEMP_RECORDS_INC Project

The EPM_TEMP_RECORDS_INC project is provided to help you integrate new temporary tables into EPM when you have added (licensed) a new EPM product but already run the Clean Temporary Tables process. Keep in mind that when you run the Clean Temporary Tables process, you delete temporary tables associated with any uninstalled EPM product. Therefore, when you introduce a new EPM product you also add new temporary tables back into the related projects. As described in this chapter, when you add new temporary tables to a project, you have to rebuild the project. The EPM_TEMP_RECORDS_INC project is an incremental project which contains only newly added temporary tables. You can use the EPM_TEMP_RECORDS_INC project to rebuild your temporary tables, instead of using the EPM_TEMP_RECORDS project with the entire set of your temporary tables. Because of its smaller size, rebuilding the EPM_TEMP_RECORDS_INC project saves you processing time.

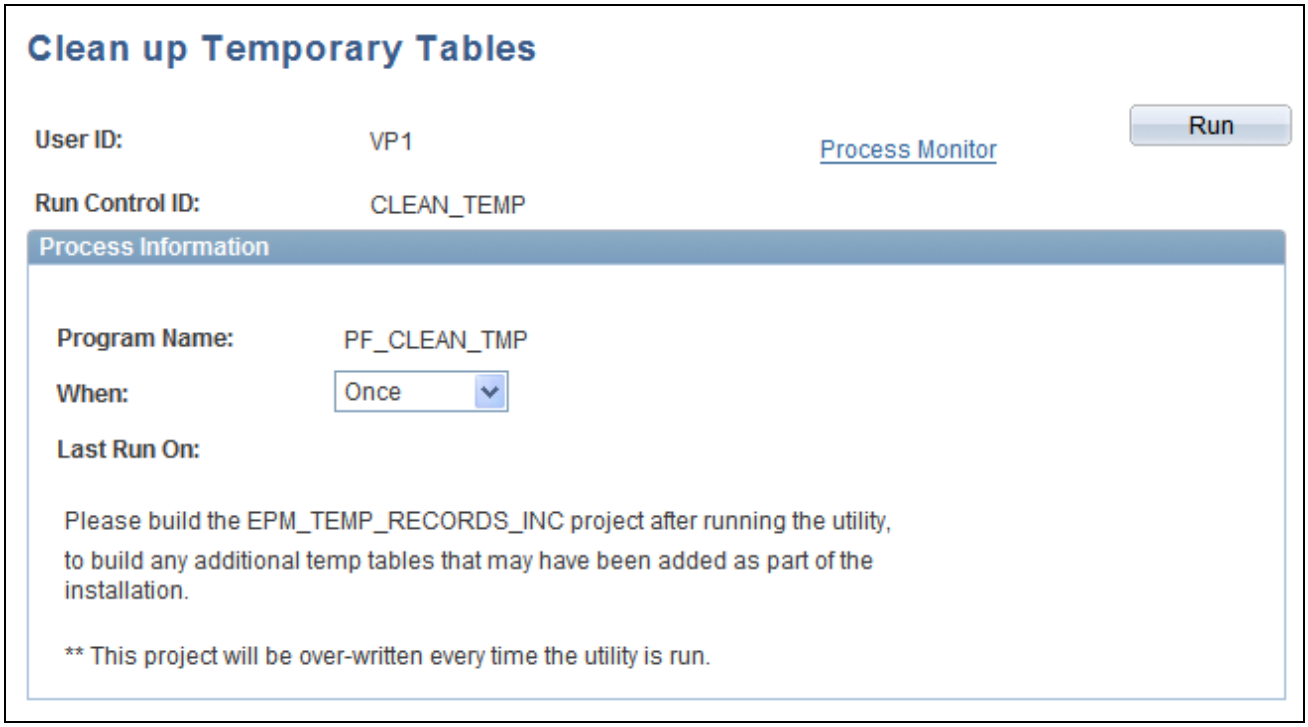
See Chapter 20, "Streamlining Processing with Jobstreams," Creating Additional Instances of Temporary Tables for Record Suites, page 481.

Pages Used to Remove Temporary Tables from EPM

Page Name	*Definition Name*	*Navigation*	*Usage*
Clean Up Temporary Tables	PF_RUN_CLEAN_TMP	EPM Foundation, Job Processing, Temporary Tables, Cleanup Temp Tables, Clean Up Temporary Tables	Run the Clean Temporary Tables application engine.
Cleanup Log	PF_TMPTBL_VW	EPM Foundation, Job Processing, Temporary Tables, Cleanup Log	View temporary tables dropped from the EPM database.

Dropping Extraneous Temporary Tables from EPM

Access the Clean Up Temporary Tables page (EPM Foundation, Job Processing, Temporary Tables, Cleanup Temp Tables, Clean Up Temporary Tables).



Clean Up Temporary Tables page

When Select the frequency in which you would like the PF_CLEAN_TMP process to run.

Run Click to run the PF_CLEAN_TMP process.

Viewing the Temporary Tables Dropped from EPM

Access the Cleanup Log page (EPM Foundation, Job Processing, Temporary Tables, Cleanup Log).

Product Name Displays the product associated with the deleted temporary tables.

DateTime Displays the date and time the process was run.

Number of temporary tables Displays the total number of temporary tables associated with the selected product.

Installed Product Indicates whether this is an installed product.

Only temporary tables from non-installed products should be dropped.

Tables Dropped Displays the total number of temporary tables dropped from the EPM database.

Table Name Displays the name of the temporary table deleted from the EPM database.

Object Owner ID Displays the Owner ID associated with a particular temporary table.

Running Jobstreams

This section discusses how to:

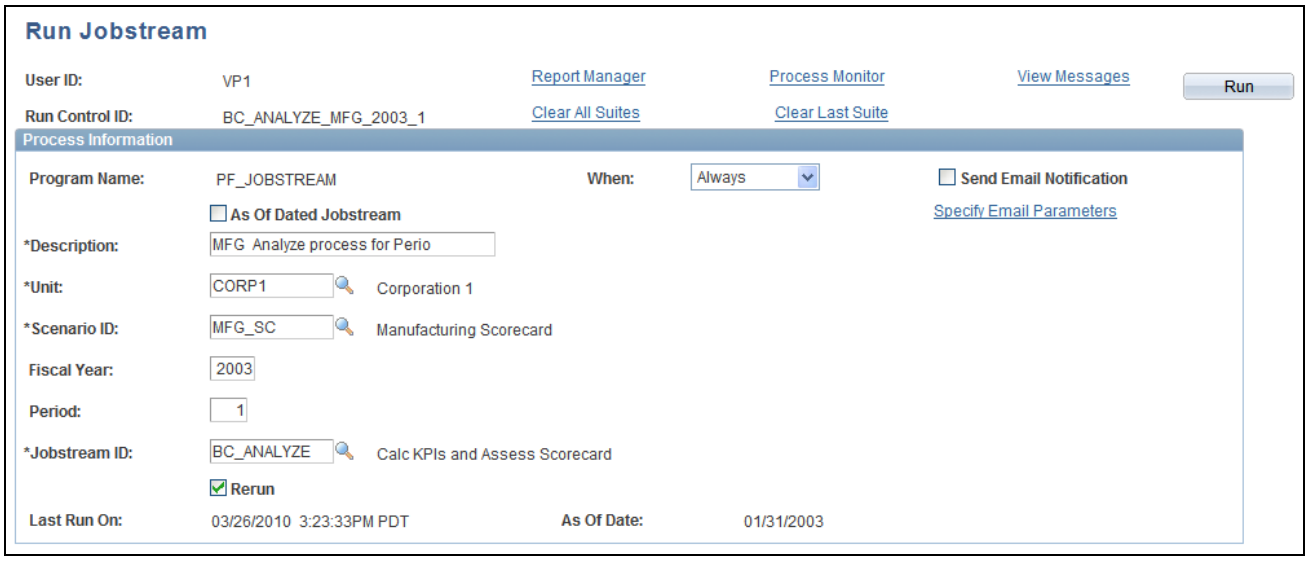
- Run jobstreams.
- Run multiple jobstreams.
- Set up email notification.

Pages Used to Run a Jobstream and Multiple Jobstreams

Page Name	*Definition Name*	*Navigation*	*Usage*
Run Jobstream	RUN_PF_JOBSTREAM	EPM Foundation, Job Processing, Update/Run Jobstreams, Run Jobstream	Run a jobstream.
Run Multiple Jobstream	RUN_PF_MULTIPERIOD	EPM Foundation, Job Processing, Update/Run Jobstreams, Run Multiple Jobstream	Run a jobstream for multiple fiscal years and accounting periods.
Jobstream Email Notification	PF_EMAIL_MSG	Click Specify Email Parameters on the Run Jobstream or Run Multiple Jobstreams page.	Set up email parameters for automatically notifying users when a jobstream is complete or abended.

Running Jobstreams

Access the Run Jobstream page (EPM Foundation, Job Processing, Update/Run Jobstreams, Run Jobstream).



Run Jobstream page

As Of Dated Jobstream Select this check box to disable the Fiscal Year and Period fields. Enter an as of date for the jobstream run.

Send Email Notification Select this check box to send an email notification to all of the email addresses that you define by clicking the Specify Email Parameters link. The email notification informs the recipients that the jobstream is complete or has abended.

Description Enter a description for the jobstream run. The Metadata Search engine uses this description to find the data later.

Unit and **Scenario ID** Select the business unit and scenario ID combination.

Fiscal Year and **Period** Enter the fiscal year and period for this jobstream run. This field does not appear if you select the As Of Dated Jobstream check box.

Jobstream ID Select the jobstream you want to run.

Rerun Select this check box if you are processing the same job an additional time using identical parameters and want the system to re-resolve the tables.

Re-resolving means that data is re-selected from the permanent table and moved to temporary tables of the assigned record suite.

Note. This option may slow down processing if you are assigned to the same record suite assigned the previous time that the engine was run.

Last Run On Displays the date and time this jobstream was last run.

As Of Date Displays the as of date for an as-of-dated jobstream. If you are using the Fiscal Year and Period, this field displays the last day of the fiscal year and period combination based on the calendars you defined.

View Messages Once a jobstream has run, click to view the engine messages generated by the jobstream. This page is described later in this chapter.

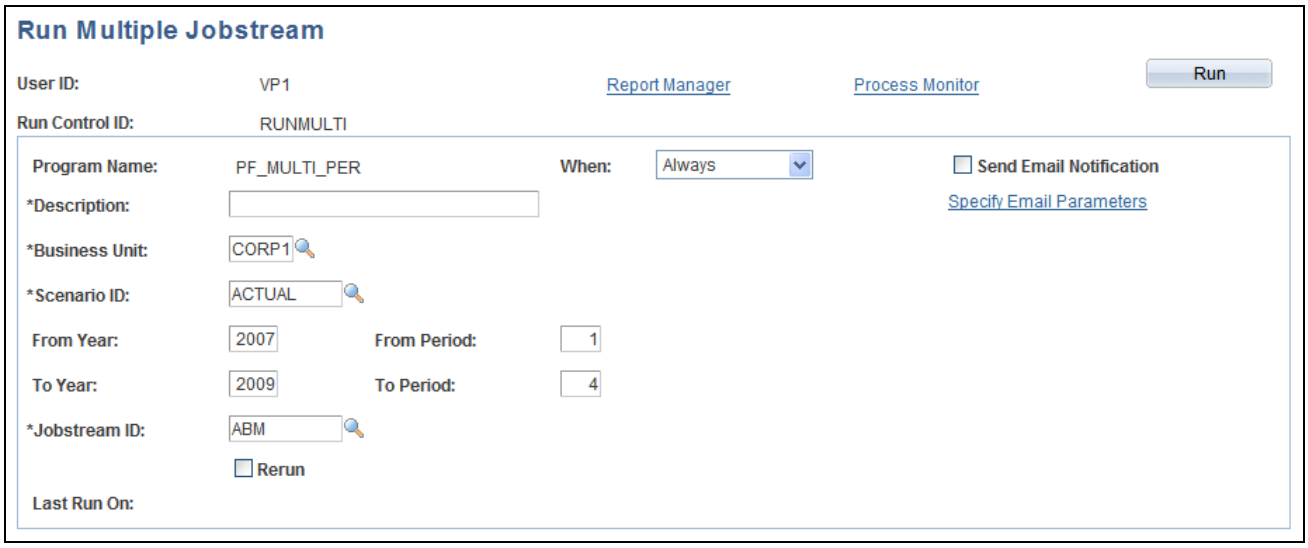
Clear Last Suite Select this option to release the last record suite used by this jobstream.

Clear All Suites Select this option to release all record suites. All record suites are now available to jobstreams.

Warning! Before clearing all record suites, make sure that no jobs are running.

Running Multiple Jobstreams

Access the Run Multiple Jobstream page (EPM Foundation, Job Processing, Update/Run Jobstreams, Run Multiple Jobstream).



Run Multiple Jobstream page

Program Name Displays the name of the jobstream program.

Send Email Notification Select this check box to send an email notification to all of the email addresses that you define by clicking the Specify Email Parameters link. The email notification informs the recipients that the jobstream is complete or has abended.

Description Enter a description for the jobstream run. The Metadata Search engine uses this description to find the data later.

Business Unit and **Scenario ID** Select the business unit and scenario ID combination.

From Year and **From Period**, **To Year** and **To Period** Enter the fiscal years and periods to include in this jobstream. Unlike the Run Jobstream page, on which you can only specify one fiscal year and period combination, you can specify a range of years and periods.

Jobstream ID Select the jobstream you want to run.

Rerun Select this check box if you are processing the same job an additional time using identical parameters and want the system to re-resolve the tables.

Re-resolving means that data is re-selected from the permanent table and moved to temporary tables of the assigned record suite.

Note. This option may slow down processing if you are assigned to the same record suite assigned the previous time that the engine was run.

Last Run On Displays the date and time this jobstream was last run.

Run Click this button to access the Process Scheduler Request page on which you define the parameters for running the jobstream.

Process Monitor Click this button to access Process Monitor pages to check process scheduler results.

Process Monitor provides updated information on the progress of reports and processes. From a Web browser, you can monitor process requests and the status of different servers that run your reports. If there are messages related to a process, you can view them from Process Monitor, as well.

Setting up Email Notification

Access the Jobstream Email Notification page (Click Specify Email Parameters on the Run Jobstream or Run Multiple Jobstreams page.).

Use this page to list the email addresses of those recipients who should receive a notification when the jobstream completes or abends. You can enter a subject for the email and any text you would like to send.

Tracking Jobs

There are a number of pages enabling you to track the progress of your jobs. This section discusses how to:

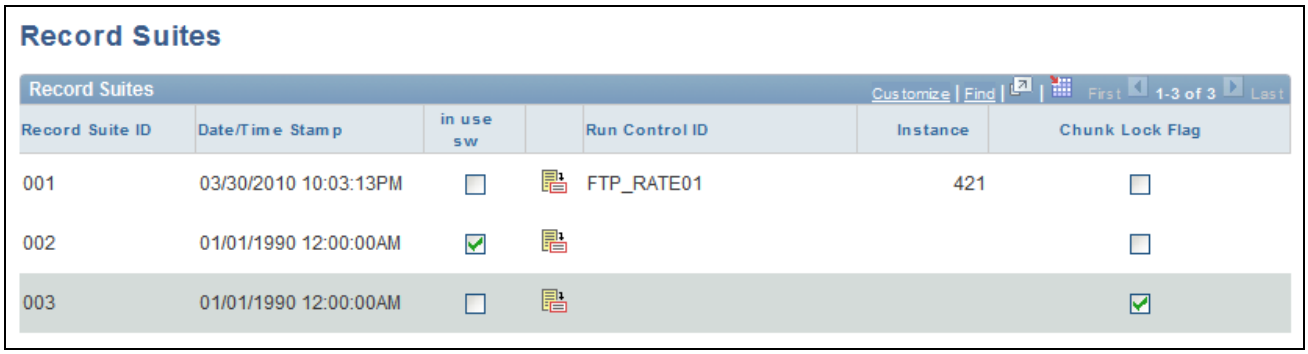
- Review record suites.
- Review record suite history.
- Review records in a jobstream.
- Review jobstream history.
- Review temporary tables.
- Review temporary table history.

Pages Used to Track Jobs

Page Name	*Definition Name*	*Navigation*	*Usage*
Record Suites	PF_RECSUITE_TBL1	EPM Foundation, Job Processing, Review Jobstream Content, Review Record Suites, Record Suites	View all of the defined record suites.
Jobstream Job Detail	PF_JOBSTRM_TBL2S	Click the  button on the Record Suites page.	View runtime parameters to determine whether a record suite is in use.
Record Suite History	PF_RECSUITE_HIS1	EPM Foundation, Job Processing, Review Jobstream Content, Record Suite History	View the process instances, job description, and run control IDs that were run in the specified record suite.
Jobstream	PF_JOBSTRM_TBL1	EPM Foundation, Job Processing, Review Jobstream Content, Review Jobstream	View the current status of a jobstream for all defined record suites.
Jobstream History	PF_JOBSTRM_HIS1	EPM Foundation, Job Processing, Review Jobstream Content, Jobstream History	View the job ID, record suites, and run control parameters that have been run for a selected jobstream.
Temporary Table	PF_TEMP_REC_TBL1	EPM Foundation, Job Processing, Temporary Tables, Temporary Table	View, for each record suite, the temporary tables that have been populated. This page also displays the run controls that were used to populate them.
Temporary Table History	PF_TEMP_REC_HIS1	EPM Foundation, Job Processing, Temporary Tables, Temp Table History	Review table usage for a record suite.

Reviewing Record Suites

Access the Record Suites page (EPM Foundation, Job Processing, Review Jobstream Content, Review Record Suites, Record Suites).



Record Suites page

Record Suite ID Lists all the record suites.

Date/Time Stamp Displays the date and time for the last or current use of this record suite.

in use sw If this check box is selected, it indicates that a record suite is being used. Deselect this check box and its related page to make the corresponding record suite available to waiting jobstreams.

Note. PeopleSoft recommends that only experienced users or the Warehouse Administrator make such a change.

Click the View Jobstream Job Details button to access the Jobstream Job Detail secondary page on which you can see all the jobstreams to which the record suite is assigned.

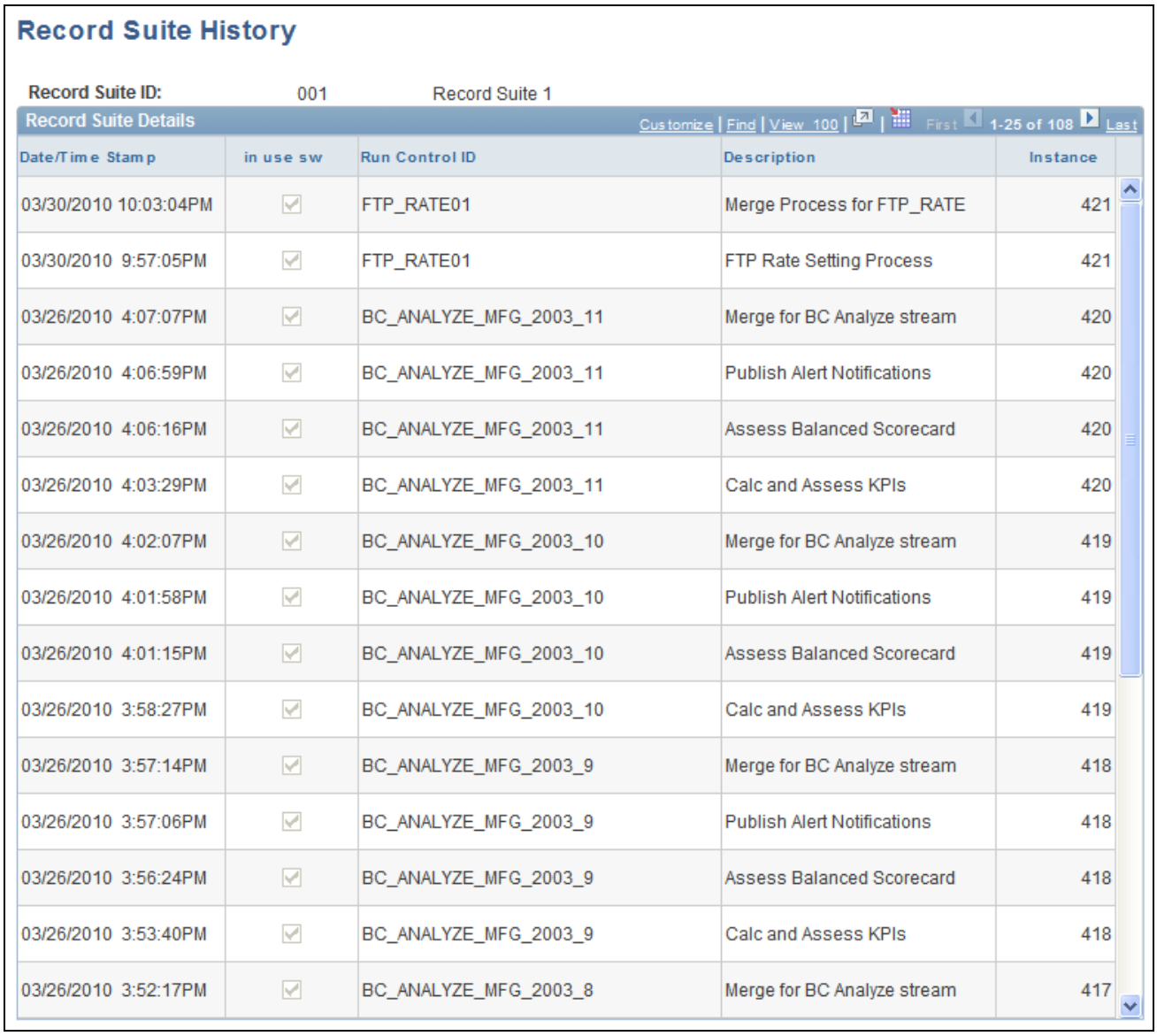
Run Control ID Displays the identifier of the run control set up to run the jobstream.

Instance Displays the process instance of the last run or current run of the run control ID.

Chunk Lock Flag If this option is selected, it indicates the record suite is in use for chunking. This check box works very much like the in use sw check box except that it is used for jobs with chunking.

Reviewing Record Suite History

Access the Record Suite History page (EPM Foundation, Job Processing, Review Jobstream Content, Record Suite History).



Record Suite History

Record Suite ID: 001 Record Suite 1

Record Suite Details — Customize | Find | View 100 | First 1-25 of 108 Last

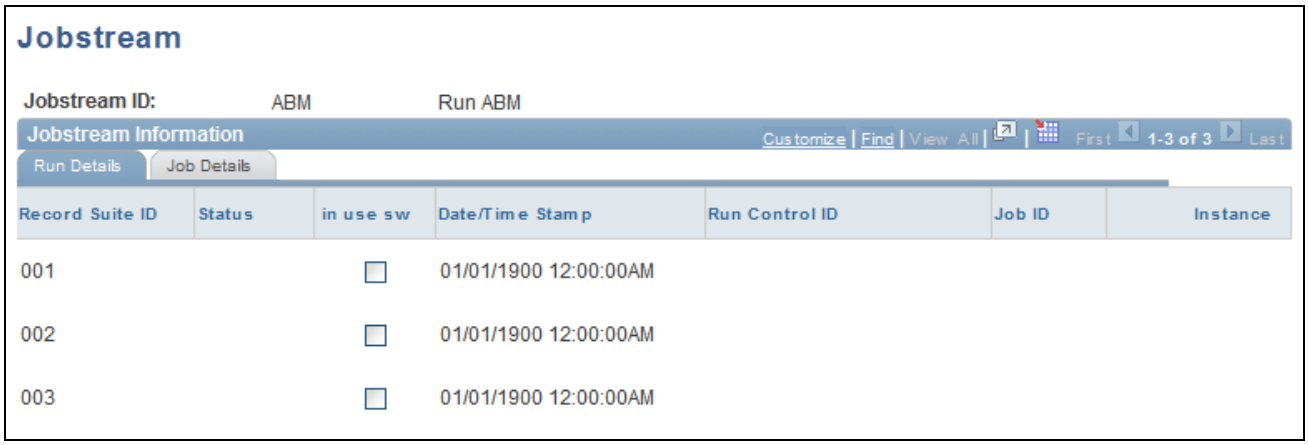
Date/Time Stamp	in use sw	Run Control ID	Description	Instance
03/30/2010 10:03:04PM	☑	FTP_RATE01	Merge Process for FTP_RATE	421
03/30/2010 9:57:05PM	☑	FTP_RATE01	FTP Rate Setting Process	421
03/26/2010 4:07:07PM	☑	BC_ANALYZE_MFG_2003_11	Merge for BC Analyze stream	420
03/26/2010 4:06:59PM	☑	BC_ANALYZE_MFG_2003_11	Publish Alert Notifications	420
03/26/2010 4:06:16PM	☑	BC_ANALYZE_MFG_2003_11	Assess Balanced Scorecard	420
03/26/2010 4:03:29PM	☑	BC_ANALYZE_MFG_2003_11	Calc and Assess KPIs	420
03/26/2010 4:02:07PM	☑	BC_ANALYZE_MFG_2003_10	Merge for BC Analyze stream	419
03/26/2010 4:01:58PM	☑	BC_ANALYZE_MFG_2003_10	Publish Alert Notifications	419
03/26/2010 4:01:15PM	☑	BC_ANALYZE_MFG_2003_10	Assess Balanced Scorecard	419
03/26/2010 3:58:27PM	☑	BC_ANALYZE_MFG_2003_10	Calc and Assess KPIs	419
03/26/2010 3:57:14PM	☑	BC_ANALYZE_MFG_2003_9	Merge for BC Analyze stream	418
03/26/2010 3:57:06PM	☑	BC_ANALYZE_MFG_2003_9	Publish Alert Notifications	418
03/26/2010 3:56:24PM	☑	BC_ANALYZE_MFG_2003_9	Assess Balanced Scorecard	418
03/26/2010 3:53:40PM	☑	BC_ANALYZE_MFG_2003_9	Calc and Assess KPIs	418
03/26/2010 3:52:17PM	☑	BC_ANALYZE_MFG_2003_8	Merge for BC Analyze stream	417

Record Suite History page

Use this page to review the process instances, job description, and run control IDs that were run in the selected record suite. A start and end time also display as well as an in use sw flag.

Reviewing Records in a Jobstream

Access the Jobstream page (EPM Foundation, Job Processing, Review Jobstream Content, Review Jobstream).

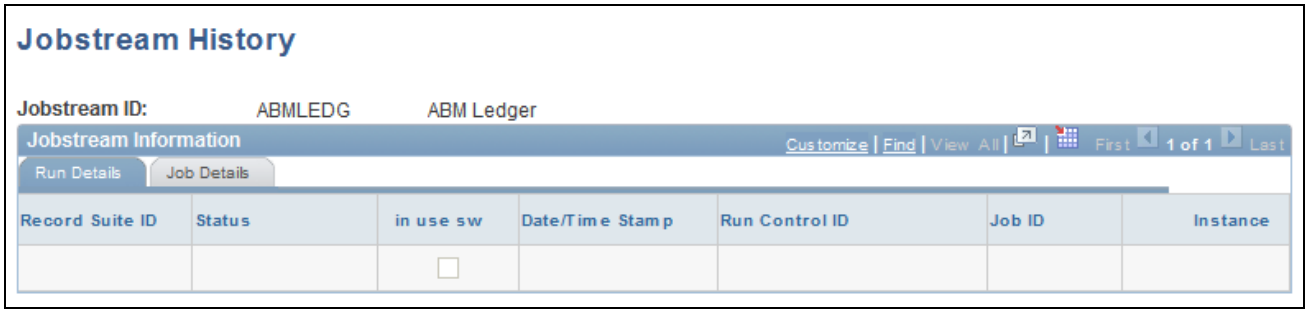


Jobstream page

Use this page to view the current status of a jobstream for all the defined record suites.

Reviewing Jobstream History

Access the Jobstream History page (EPM Foundation, Job Processing, Review Jobstream Content, Jobstream History).

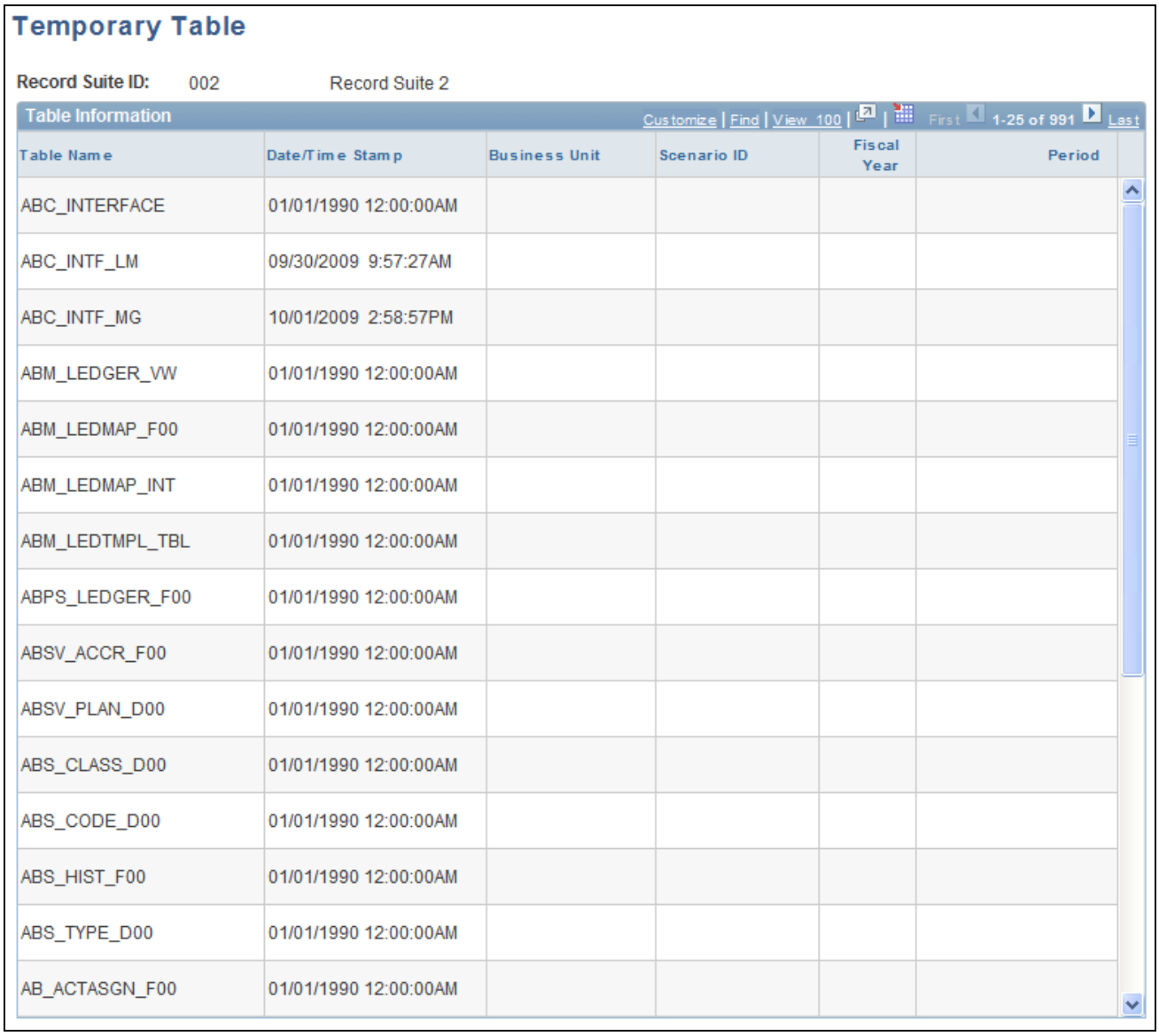


Jobstream History page

Use this page to view the job ID, record suites, and run control parameters that have been run for a jobstream.

Reviewing Temporary Tables

Access the Temporary Table page (EPM Foundation, Job Processing, Temporary Tables, Temporary Table).

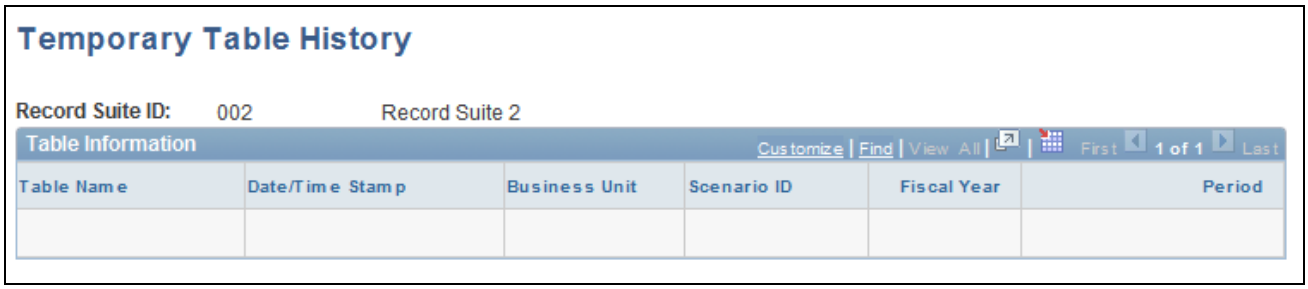


Temporary Table page

Use this page to view for a given record suite the temporary tables that have been populated. This page also shows the run control parameters.

Reviewing Temporary Table History

Access the Temporary Table History page (EPM Foundation, Job Processing, Temporary Tables, Temp Table History).



Temporary Table History page

Use this page to review table usage for a record suite.

Viewing Engine Messages

After you run a job or jobstream, view the process information and run control parameters for the engine that you just ran using the Messages component. You can access this component directly from the Run Jobstream page by clicking the View Engine Messages link.

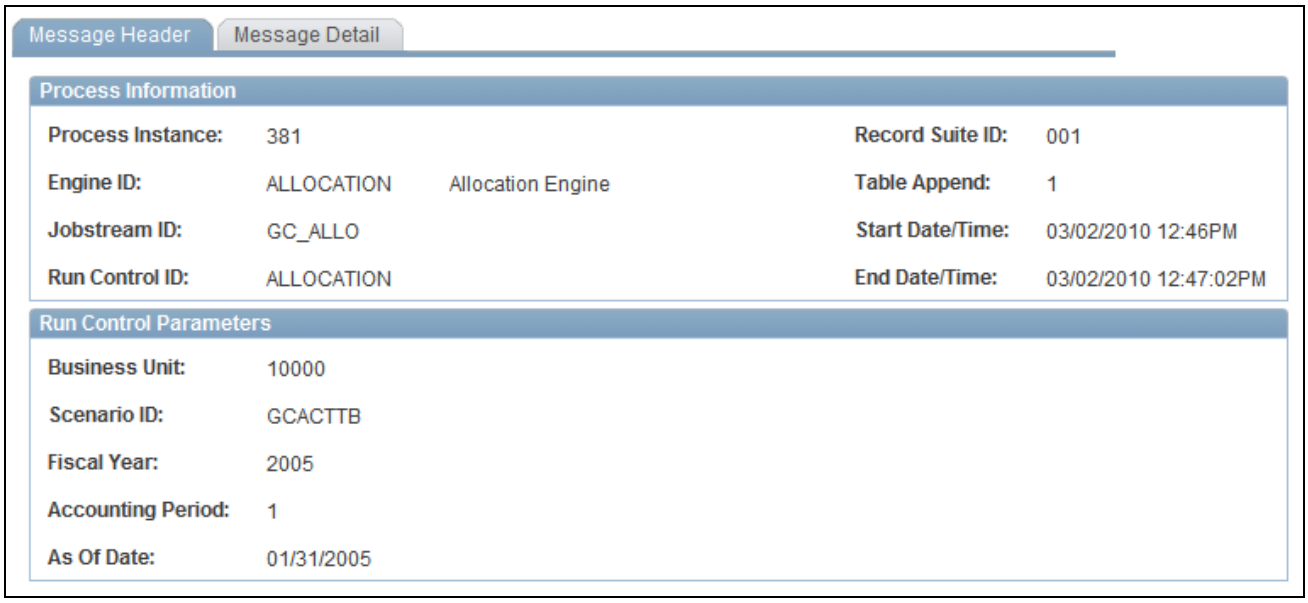
This section discusses how to view engine messages

Pages Used to View Engine Messages

Page Name	*Definition Name*	*Navigation*	*Usage*
Engine Messages - Message Header	PF_ENGMSG_HEAD	EPM Foundation, Job Processing, Review Jobstream Content, Engine Messages, Message Header	View display-only process information such as record suite ID and engine ID, as well as the run control parameters for this process instance.
Engine Messages - Message Detail	PF_ENGMSG_LOG	EPM Foundation, Job Processing, Review Jobstream Content, Engine Messages, Message Detail	View display-only process information such as source name, field name, and field value.
Message Detail	PF_ENGMSG_MSG	Click the  on the Message Detail page.	View the detailed error message.

Viewing Engine Messages

Access the Message Header page (EPM Foundation, Job Processing, Review Jobstream Content, Engine Messages, Message Header).



Message Header page

Process Information View details such as the process instance, record suite, engine ID, jobstream ID, run control, table append, as well as the start and end date and time.

Run Control Parameters Displays the defined run control parameters for the instance (business unit, scenario ID, fiscal year and accounting period, and as of date if applicable).

Go to the next page in the component to view engine message details.

Viewing Message Details

Access the Engine Messages - Message Detail page (EPM Foundation, Job Processing, Review Jobstream Content, Engine Messages, Message Detail).

For a given process instance, engine ID, jobstream ID and run control you can view the message details.

Search Click this button to retrieve engine message details. The message details appear in the grid at the bottom of the page.

Engine Message Details Displays the source of the error. In the message box, you can view the message set to which the message belongs, the message number, and a brief description of the error.

 Click the Explain First button to access the Message Detail secondary page on which you can view a detailed error message for the error.

Chapter 21

Setting Up and Using Profit Manager

This chapter provides an overview of the profit manager tools and discusses how to:

- Set up Ledger Mapper.
- Create error log reports for the performance ledger.
- Create, edit, and approve journal entries manually.
- Process and post journals.
- Use balancing and reconciliation features.
- Use ledger drill down.
- Review error messages.
- Correct profit manager fact table errors using PF Modification.
- Correct OWE dimensions.

Understanding Profit Manager Tools

The profit manager is a set of integrated tools that enable true multidimensional profitability reporting. To obtain true and meaningful profitability reports, you need a central repository as well as reliable and consistent data, and you need to consolidate and enrich the data from your general ledger and other sources.

PeopleSoft EPM infrastructure is the underlying framework that provides reliable and consistent data and consolidates data sources such as your general ledger and the EPM Analytical Applications such as PeopleSoft Activity-Based Management, Enterprise Scorecard, Funds Transfer Pricing, Risk-Weighted Capital, Global Consolidations, and Workforce Analytics, which are application engines that enrich and transform data.

The features that are described in this section are tightly integrated with the PeopleSoft analytical applications and provide you with ways to:

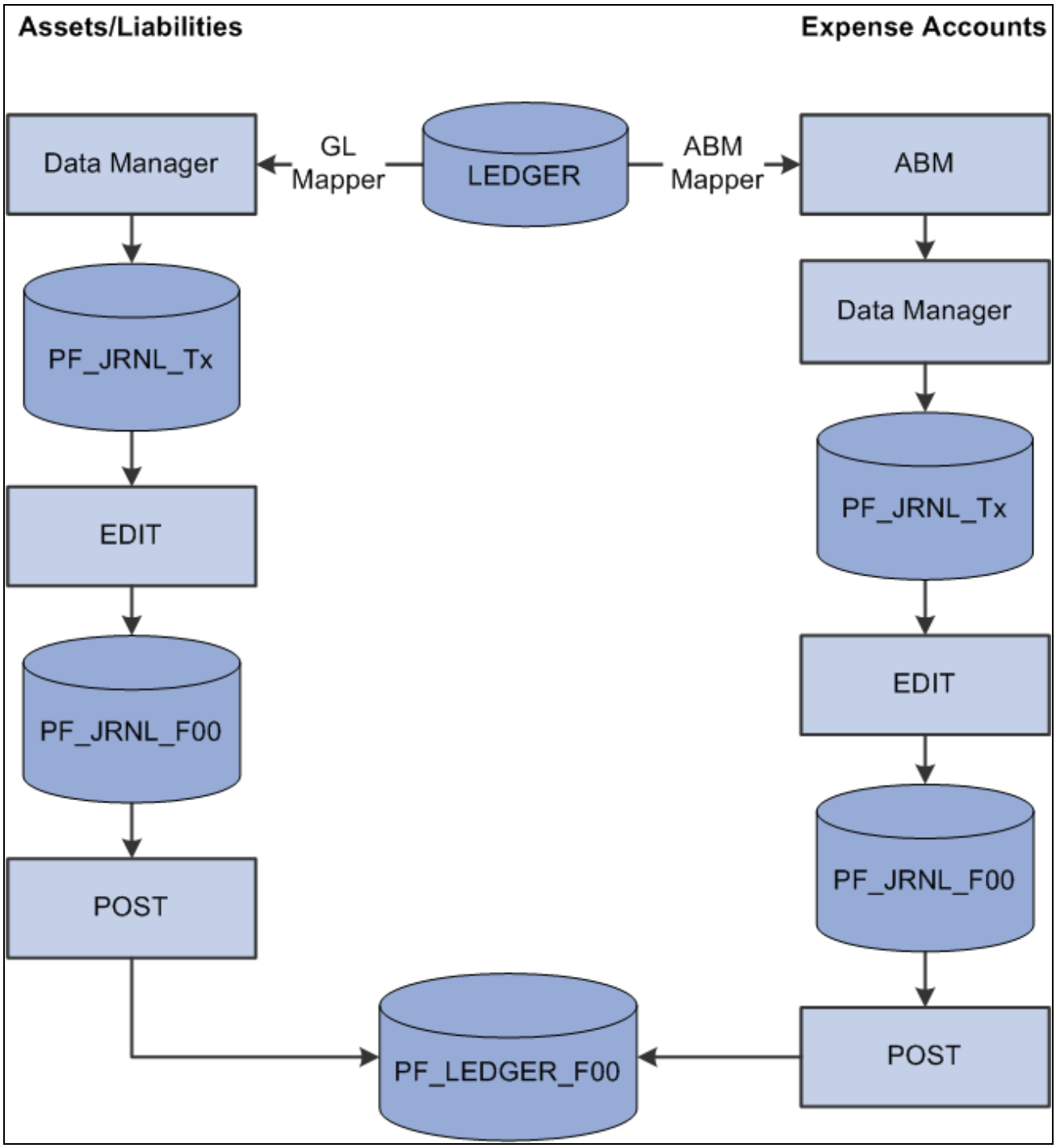
- Move data from your PeopleSoft general ledger using the Ledger Mapper.
- Verify the accuracy of your data before you post it to the performance ledger table using the PF Edit engine.
- Track data movement to and from the performance ledger table (PF_LEDGER_F00) using ledger drill down.
- Control batch processing using the PF Post and PF Unpost engines.

- Keep the contents of the journal table (PF_JRNL_F00) clean using the PF Journal Cleanup engine.
- Reconcile final table amounts using the balancing and reconciliation features.

After you set up and run a source engine or map ledger balances using the Ledger Mapper, the enriched data is moved to the performance journal table (PF_JRNL_F00). You can run the PF Journal Edit engine to check data integrity at any time. Use PF Journal Modification to revise errors. The PF Ledger Post process moves your data from the performance journal table to the performance ledger table (PF_LEDGER_F00) for reporting.

To check that data migration and enrichment is accurate:

1. Use the PF ledger drill down feature to track the source of general ledger data for a particular performance ledger after you populate the performance ledger table.
2. Use the Reconciliation utility to check balances between tables such as GL_LEDGER and PF_LEDGER.



Profit Manager loading the performance ledger table

Ledger Mapper

Use the Ledger Mapper to map data, such as assets and liabilities, that does not enter the system through one of the optional analytical applications. After you define the ledger mapping rules, you must set up Data Manager rules using the GL mapper method and then run the Data Manager or Allocation Manager engine to populate the performance journal table (PF_JRNL_F00).

For example, PeopleSoft uses the Ledger Mapper to map expense data from the general ledger to Activity-Based Management resources.

To map ledger amounts, you:

1. Load the GL_LEDGER table.
2. Use the general ledger mapper method in Data Manager to move the general ledger data into the GLSTG temporary table that you identify as the source table for further data movement.

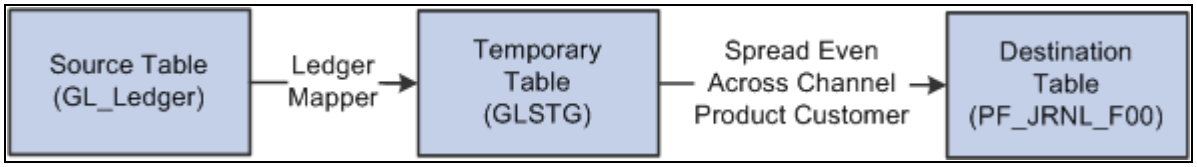
   Note. You might decide to move data directly into the performance journal table (PF_JRNL_F00) using the copy method. You can also create multidimensional data using the prorata or spread even methods.

Note. A number of steps in this process use the Data Manager or the Allocation Manager engine. You must define tablemaps, datamaps, constraints, and filters, and then set up the Data Manager or Allocation Manager rules that use this metadata to produce multidimensional results. In addition, you must set up job metadata and jobstreams.

The following two diagrams illustrate how Ledger Mapper moves the data. The first diagram illustrates Ledger Mapper and the copy method:

Using the Ledger Mapper and copy methods to load the performance journal table (PF_JRNL_F00)

The second diagram illustrates a method in which multidimensional data is created:



Using the Ledger Mapper to move ledger data, create multidimensional data, and load the performance journal table (PF_JRNL_F00)

The Ledger Mapper uses the business unit relationships that you established in EPM to map amounts. To map multiple general ledger accounts to one performance account, set up ledger mapping rules on the Ledger Mapper page.

Note. You must define your general ledger and warehouse business units and the relationship between them before you set up and run the Ledger Mapper. In addition, before you map ledgers, you must define the tablemaps, datamaps, and constraints to use when you run the Data Manager or Allocation Manager engines. EPM is delivered with predefined tablemaps, datamaps, constraints, and Ledger Mapper Data Manager rules for the SHARE setID.

See Chapter 14, "Importing Source Business Units into EPM to Create Warehouse Business Units," page 283 and Chapter 16, "Setting Up and Working with Metadata for the Operational Warehouse - Enriched," page 339.

Setting Up Ledger Mapper

This section lists prerequisites and discusses how to:

- Map ledger data.
- Create data manager rules for the general ledger (GL) mapper method.
- Configure ledger mappings for new or configured ChartFields.

Prerequisites

Before you can set up your Ledger Mapper, you must:

- Complete your warehouse business unit setup.
- Specify your ledger mapping defaults.
- Set up your tablemaps, datamaps, filters, and constraints.

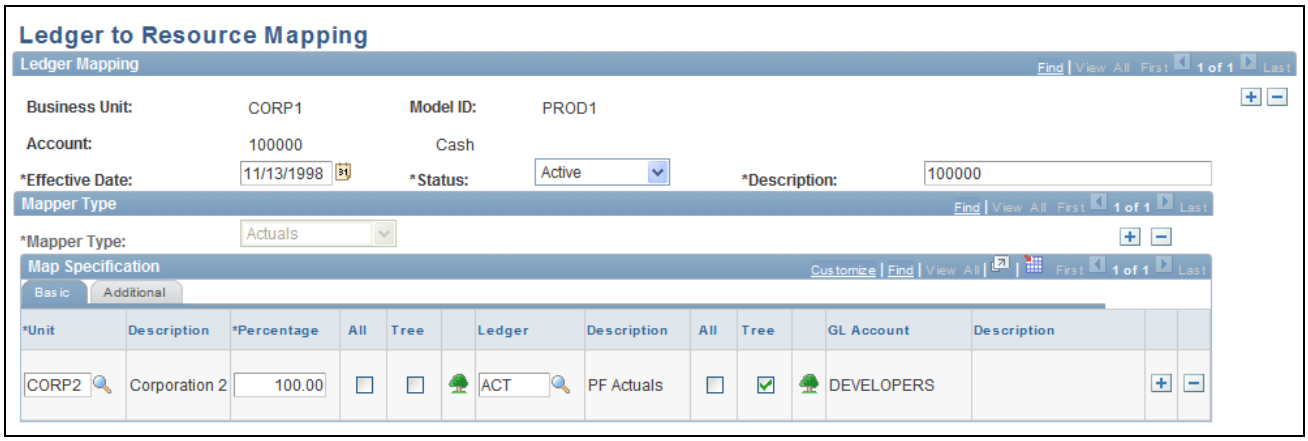
See Chapter 14, "Importing Source Business Units into EPM to Create Warehouse Business Units," page 283; Chapter 18, "Setting Up Business Rules for the Operational Warehouse - Enriched," Specifying Ledger Mapping Defaults, page 425 and Chapter 16, "Setting Up and Working with Metadata for the Operational Warehouse - Enriched," page 339.

Page Used to Set Up the Ledger Mapper

Page Name	*Definition Name*	*Navigation*	*Usage*
Ledger to Resource Mapping	LEDMAP_TBL1	EPM Foundation, Data Enrichment Tools, Performance Ledger, Map Ledger Data, Ledger to Resource Mapping	Map ledger data, such as assets and liabilities, that do not enter the system through one of the analytic applications.

Mapping Ledger Data

Access the Ledger to Resource Mapping page (EPM Foundation, Data Enrichment Tools, Performance Ledger, Map Ledger Data, Ledger to Resource Mapping).



Ledger to Resource Mapping page

Mapper Type Specify *Actuals* or *Budgeted* for mapper type.

This field is for the Activity-Based Management analytical application only.

Unit Select the general ledger business unit to which you are mapping from the prompt list. Depending on your ledger mapping default setup, you might not have to enter information for the rest of the delivered ChartFields.

Description Displays the description of the general ledger business unit.

Percentage Use this field to split the ledger data from one general ledger business unit into multiple performance accounts or resources in Activity-Based Management. Do this by specifying that only a certain percentage from a ledger gets mapped.

For each delivered ChartField (ledger, general ledger account, operating unit, department, and product), enter the following information:

Description Displays the description of each delivered ChartField value or tree node to include in the mapping.

All Select this check box to bring all valid values for the specified ChartField (for example, to bring all department data to the performance account).

Tree Select this check box to use values from a tree. If the *Tree Name not Defined* error appears, you must specify the tree name on the Ledger Mapping Defaults page. To create a new tree, use the PeopleTools Tree Manager.

 Click the Tree button to specify the tree name and the tree node.

See Also

Chapter 18, "Setting Up Business Rules for the Operational Warehouse - Enriched," Specifying Ledger Mapping Defaults, page 425

Creating Data Manager Rules for the GL Mapper Method

You can now set up the Data Manager rules. You must set up a Data Manager rule for each setID for which you want to map ledger data. The data manager rule creates the SQL code for that setID.

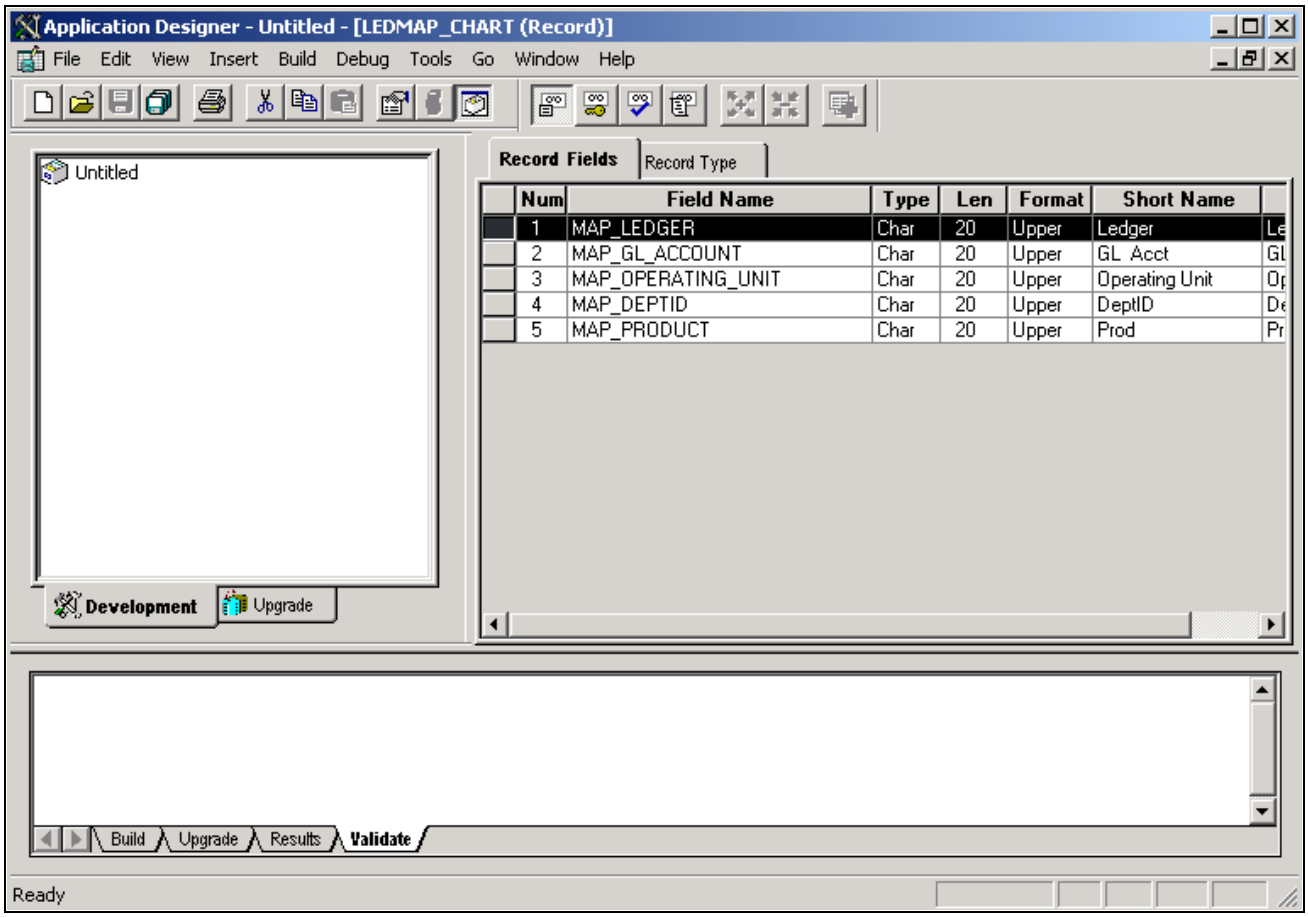
See Also

Chapter 22, "Using Data Enrichment Tools," Defining Data Manager Rules, page 551

Configuring Ledger Mappings for New or Configured ChartFields

This section describes how to configure ledger mapping with any configured or new ChartFields that you want to use with Enterprise Performance Management. This process ensures that the ChartFields appear on the Ledger Mapping Defaults pages and other Ledger Mapper pages.

Warning! Be sure to save your original settings before configuring. Before you attempt any configurations, you should be familiar with the PeopleTools Application Designer and Application Engine, and your ChartField structure.



Adding ChartFields to the LEDMAP_CHART subrecord

To configure ledger mappings:

1. Add the ChartFields to the subrecord LEDMAP_CHART.

   To open the record, launch the PeopleTools Application Designer and select File, Open. Open the LEDMAP_CHART subrecord and add the new ChartFields. Fields should be MAP_[*name*]. Make the new fields key fields.

   Add ChartFields depending on whether a prompt table exists. When you are finished, save the record. Add the new ChartField to the LED_KEY2_SBR record and make it a key field. Also add the new ChartField to the LED_KNK_SBR record, but do not designate it as a key.

2. Add the ChartFields to the SQL object LEDMAP_CHART.

   To open the SQL object, select File, Open in the Application Designer. Open the LEDMAP_CHART SQL object (object type: SQL), and add the new ChartFields. Fields should be MAP_[*name*].

3. Alter the corresponding tables and views for each of the modified subrecords.

   You should alter them in the following order: LED_KNK_SBR, LED_KEY2_SBR, LEDMAP_CHART.

   Note. You must use Find Object References to determine which tables and views need to be altered.

4. Alter the LEDMAP_SEQ grid on the page LEDMAP_TBL1.

   In the Application Designer, access the page LEDMAP_TBL1. Go to the LEDMAP_SEQ grid and add or update the following fields:

Detail	***Description***
Check box CHARTFIELD_X_OPT*	Record DERIVED_LEDMAP (Short description) (Yes/No)
Edit box CHARTFIELD_X_OPT	Record LEDMAP_WORK (Long description, Invisible)
Check box CHARTFIELD_X_TOPT	Record LEDMAP_SEQ (Short description) (Yes/No)
Edit box CHARTFIELD_X_TREE	Record DERIVED_LEDMAP (Long description, Invisible)
Push Button CHARTFIELD_X_TSEL	Record DERIVED_LEDMAP (copy image from previous ChartField)
Edit box MAP_[*name*]	Record LEDMAP_SEQ (Long description)

Note. *X* refers to the ChartField number. If you insert the new ChartField between existing ChartFields on the LEDMAP_CHART subrecord, you must add the new fields and renumber the existing fields. The subrecord sequence number must match the CHARTFIELD_X_OPT.

Note. Before saving the grid and page, adjust the size of all boxes for All/Tree/Tree Name for ChartField.

5. Add the following fields to the PF_LED_TMPL_TBL1 page:

Detail	*Description*
Edit box FIELDNAMEX	Record DERIVED_LEDMAP (None, Invisible, Display Control) Label Text = "FIELDNAMEX"
Edit box CHARTFLD_X_TREEVW	Record PF_LED_TMPL_TBL (None)
Edit box CHARTFLD_X_NODEVW	Record PF_LED_TMPL_TBL (None)
Edit box SHORTNAME	Record PSDBFIELD (None, Display-Only, Related Display) Label Text = " FIELDNAMEX descr"

Note. *X* refers to the ChartField number. When you save the data, you can ignore the warning message *PSDBFLDLABL. SHORTNAME occurs more than once.*

If you receive any errors, check the layout order and reorder the fields as they should appear on the page.

6. Add the GL mapper rule for the current setID on the Data Manager Rules Define Rule page if it is not there already.

Warning! If the ABMP rule already exists for the current setID, you *must* rebuild the SQL by resolving the page.

7. To use trees, specify the appropriate tree view, tree node view, and tree name on the Ledger Mapping Defaults page.

   Set up tree view and tree node views for the new ChartFields. Also, be sure to add the views under a record group control for the ChartField (add a new record group if one does not exist). Access the Ledger Mapper Defaults II page and set up a tree name if needed.

8. Correct existing ledger to resource mappings to reflect the new ChartField.

   Use the All Values option as a default or specify a value or tree.

Creating Error Log Reports for the Performance Ledger

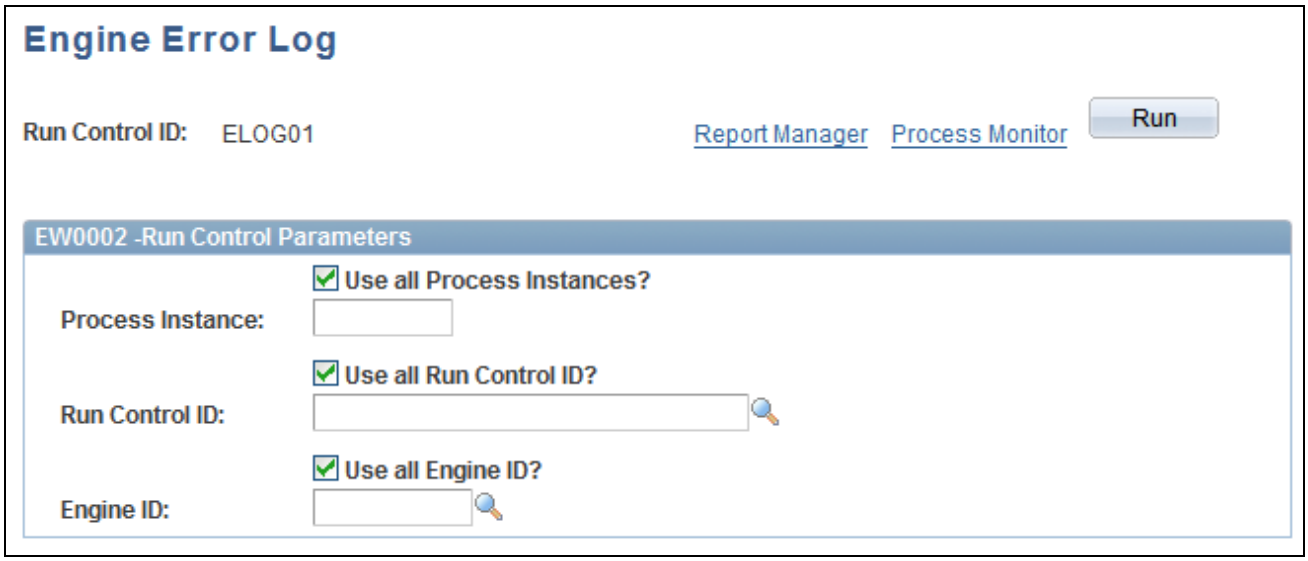
You should create an engine error log report to ensure that you have successfully mapped the data. The error log is generated during the Ledger Mapper process.

Page Use to Create Error Log Reports

Page Name	*Definition Name*	*Navigation*	*Usage*
Engine Error Log	RUN_REW_0002	EPM Foundation, Data Enrichment Tools, Performance Ledger, Create Error Log Report, Engine Error Log	Create error logs for PeopleSoft application engine processes.

Creating Error Logs

Access the Engine Error Log page (EPM Foundation, Data Enrichment Tools, Performance Ledger, Create Error Log Report, Engine Error Log).



Engine Error Log page

Process Instance and **Use all Process Instances?** Specify a specific process instance or create the error log for all instances.

Run Control ID and **Use all Run Control IDs?** Specify a run control or create the error log for all run controls.

Engine ID and **Use all Engine IDs?** Specify a particular engine or create the error log for all engines.

Click Run to define parameters for running the Engine Error Log (EWC002) Crystal report.

Creating, Editing, and Approving Manual Journals

This section provides overviews of manual journals, the journal copy feature, and reverse journal entries, and discusses how to:

- Enter manual journals.
- Copy journals.
- Create reverse journal entries.
- Attach supporting documentation for manual journal entries.
- Approve journals.

Understanding Manual Journals

Before you post your journals, you can use the Journal Entries page to manually record transactions and create journal debit and credit entries. Journals are written to a journal fact table (such as GC_JRNL_MGT_F00), which is the source record for the various ledger posting processes).

Warning! Before you post a journal, ensure that you have properly mapped the Foreign Currency Code in the Data Manager - Define Target page.

If you do not populate the Foreign Currency Code, unexpected results may occur when the PF_POST application engine runs.

See Chapter 22, "Using Data Enrichment Tools," Defining the Target, page 561.

Understanding the Journal Copy Feature

You can copy a manual journal to:

- Record the same journal entry to multiple scenarios.

  The journal must use the common consolidation business unit.

- Duplicate a journal for another fiscal year or period.
- Create reversal journal entries.

You can copy any existing valid or posted journal that was created with or copied from the Manual Journal Entries page.

If the base currency of the target journal's scenario is different from that of the source journal, the system calculates the new base amounts by:

1. Retrieving the currency exchange rate from the transaction currency and converting it to the new base currency, using the rate type that you specify on the line and the journal date as the currency effective date.

   Rate type is required on all manual journals.

2. Computing the new base amount.

   The new base currency amount is equal to the original transaction amount divided by the product of the rate divisor and rate multiplier that is applicable for currency conversion, rounded to the number of decimals that are specified for that base currency.

When copied, if the journal is out of balance because of rounding from any required currency conversion, the system adjusts the base amount of the first journal line.

Regardless of the source journal system source, the target journal system source is set to SCG, which means that the journal was system-generated and originated from the Journal Copy process.

Understanding Reverse Journal Entries

The Journal Reversal page enables you to create a reverse journal entry or reversal batch. The reversal batch is created in a valid (edited) status that does not require journal edits. Alternatively, you can copy your original journal, select the reverse amount check box, and designate the journal date to create a reversal journal.

Reverse journals are associated with your original journal by the journal ID. If you try to post a reversal journal to the system before posting the original, you receive an error message that the original journal has not yet been posted and the reversal does not post. If you should attempt to unpost the original journal that is associated with the reverse journal, you receive an error message specifying that this action cannot be performed. Similarly, if you attempt to unpost a reversal journal entry, you receive a warning saying that you must also unpost the original after unposting the reversal.

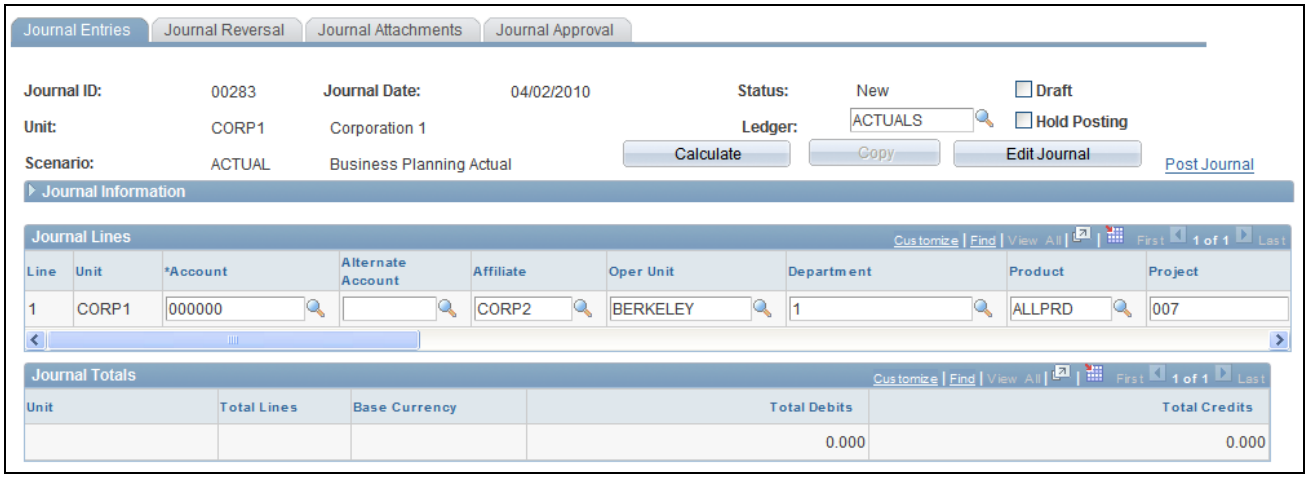
Pages Used to Create, Edit, and Approve Manual Journals

Page Name	*Definition Name*	*Navigation*	*Usage*
Journal Entries	PF_JOURNAL_TBL1	EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Journal Entries, Journal Entries	Enter or copy journals.
Journal Copy Options	PF_JOURNAL_CPY	Click the Copy button on the Journal Entries page.	Create journal copies.
Journal Reversal	PF_JOURNAL_TBL3	EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Journal Entries, Journal Reversal	Select the option to create a reversal batch and the date of the reversal journal entries.

Page Name	Definition Name	Navigation	Usage
Journal Attachments	PF_JRNL_ATTACHMENT	EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Journal Entries, Journal Attachments	Attach documentation to support journal entries.
Journal Approval	PF_JOURNAL_TBL2	EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Journal Entries, Journal Approval	Set the approval status for a journal.

Entering Manual Journals

Access the Journal Entries page (EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Journal Entries, Journal Entries).



Journal Entries page

Ledger ID Select a ledger ID which is used to post the journal to the ledger.

Draft Select this check box to create a draft journal. This option excludes the journal from the manual journal edit process.

Hold Posting Select this check box to prevent the journal entry from posting during ledger post processing.

This option enables you to save the journal entry and continue to add information to it at a later time.

Calculate Click to initiate the calculate process and update the totals in the Journal Information region and Journal Totals grid.

For each journal line on the journal line grid, the system compares all field values with the prior values that are stored in the buffer for any changes. If changes exist on currency, rate type, exchange rate, or base currency, the system determines the new exchange rate. This process also initiates exchange rate checking before the new exchange rate can be accepted. The new exchange rate is then converted to rate multiplier and rate divisor values. If any changes occur in amount, exchange rate, or base amount, the system calculates the unchanged amount based on the system rule or the rule that you specify in the calculate field. The system then adjusts the journal totals and balancing ChartField totals to reflect the new totals. Even if no amount field change occurs but a ChartField value change occurs, the system adjusts the balancing ChartFields totals accordingly.

Copy Click to access the Journal Copy Options page and create a copy of this journal.

Note. This option is not available until you save the journal.

Edit Journal Click to edit (validate) your journal.

Your journals cannot be posted until they are valid. If errors occurred on the journal, you can correct the errors with the Ledger Edit component.

Post Journal Click this link to access the Post Ledger run control page to post the journal.

Note. This option is not available until the journal is valid (edited).

Journal Information

Expand the Journal Information region to complete or review these fields:

Journal Source Code Select the source code for the journal.

Journal source codes enable you to track the source of your journal data. Source codes are defined with the Journal Source Code page.

Note. This is an optional field.

The system derives values in the Fiscal Year and Period fields from the scenario's calendar, based on the journal date. The system derives the value in the Base Currency field from the business unit definition. The amounts for total debits and total credits for the entire journal appear in the Journal Totals grid.

Journal Lines

Add rows to the Journal Lines grid to enter additional journal lines. For each line, you must complete the Account and Amount fields. If the transaction currency is not the same as the base currency, you must also specify values for the Currency and Rate Type fields. The system uses the associated exchange rate to populate the base amount. The consolidation dimension field label varies depending on the consolidation dimension. For example, if the consolidation dimension is business unit, the field label is Ledger Unit. The ledger template that is associated with the business unit for the journal entry determines the ChartFields that are available for the journal entry. You can use multiple business units within the same ledger ID.

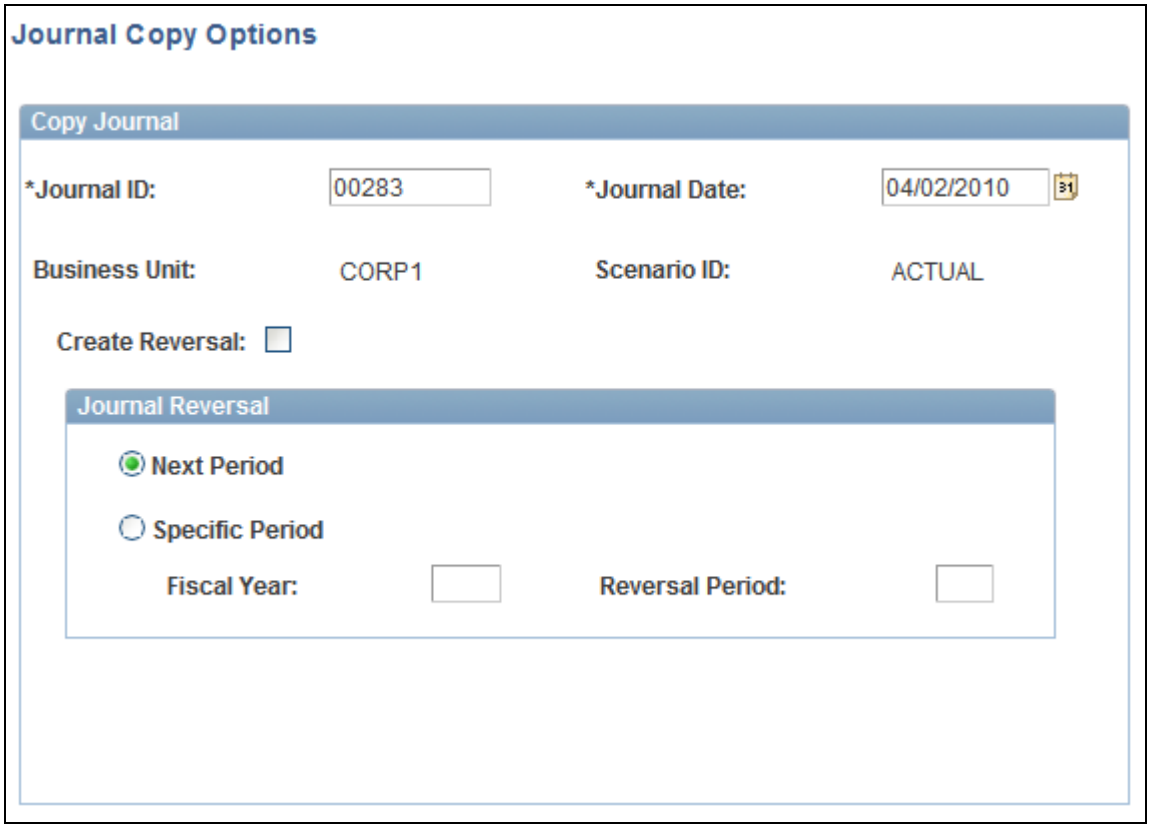
Note. You can customize the ChartFields that appear by clicking the Customize link on the grid.

Journal Totals

Use the Journal Totals grid to view a summary of the journal entry. The information is grouped by dimension value (such as ledger unit) and lists the total number of lines and total debits and credits.

Copying Journals

Access the Journal Copy Options page (Click Copy on the Journal Entries page).



Journal Copy Options page

To copy a journal:

1. Access the Manual Journal Entries page, select the journal that you want to copy, and then click Copy.

   The Journal Copy Options page appears. The Copy Journal region lists the values that you are copying from the source journal.

2. Complete the fields in the Copy Journal group box:

   Journal ID Enter a journal ID for the new journal.

   The default value in this field is the same value as the journal that you are copying; however, you can override this value.

   Journal Date Enter a date for the new journal.

   If the target journal date is different from the source journal date, the system uses the target journal date to derive the fiscal year and accounting period values.

   Create Reversal Select this check box to create a reversal batch for this journal.

   If you select this option, specify the period in which you want the reversing journal entry to occur. You can specify the next period based on your original journal date or by specifying a specific period in time.

3. Click OK to copy the journal.

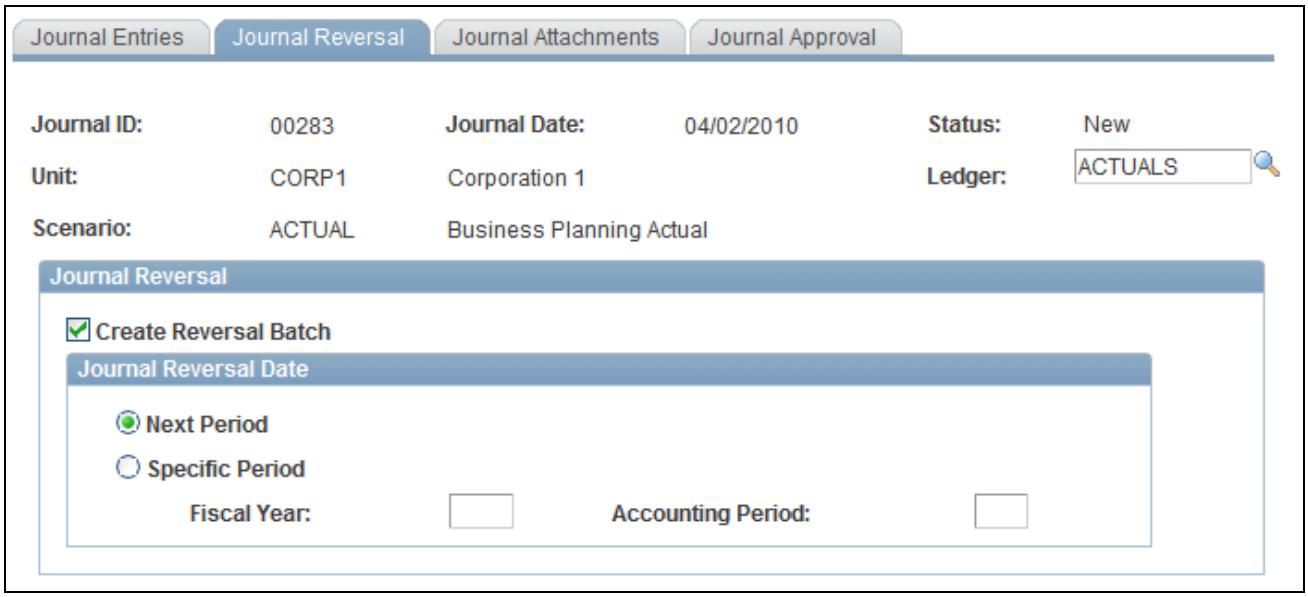
   The Journal Copied page summarizes the scenarios to which the journal was copied. Any copied journals that require approval (through workflow or security) have an initial approval status of *None*.

4. Review the journal and click Submit for Approval to initiate the approval process.

   For preapproved journals, the approval status is immediately set to *Approved*, which copies the journal lines to the journal fact table that you specified on the ledger template.

Creating Reverse Journal Entries

Access the Journal Reversal page (EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Journal Entries, Journal Reversal).



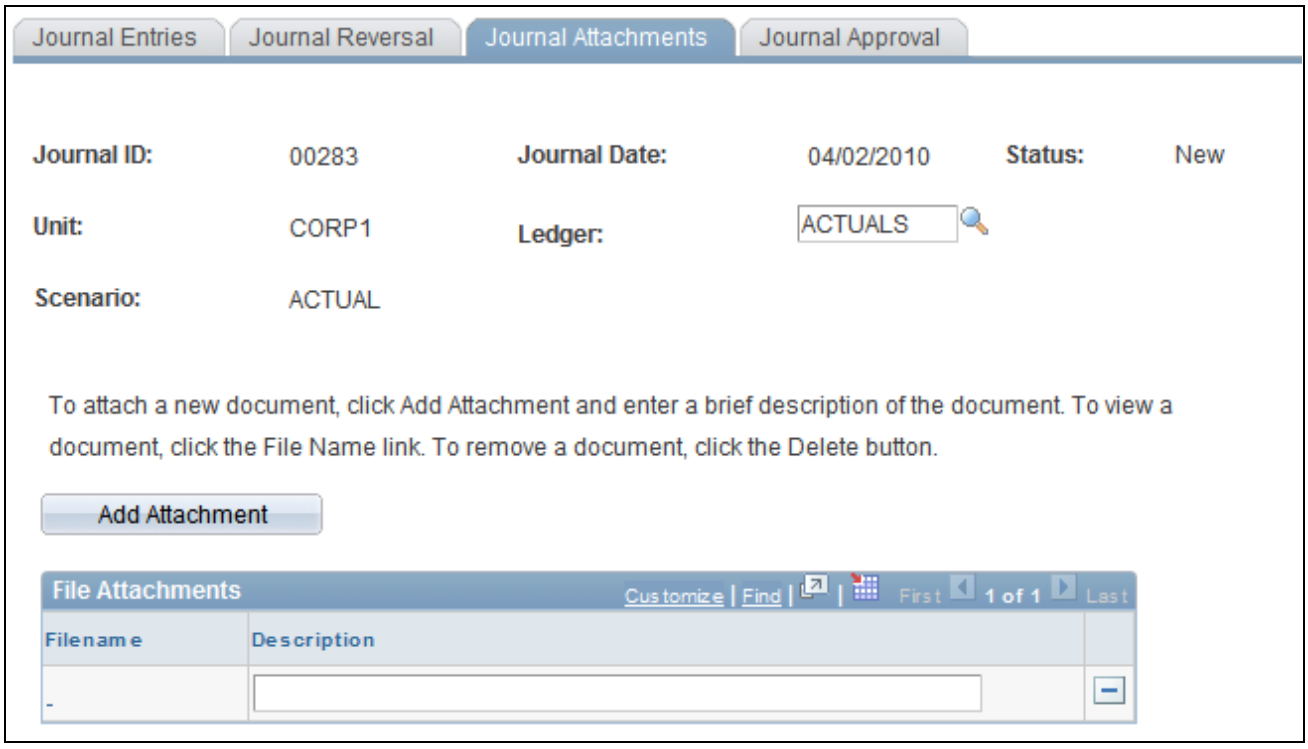
Journal Reversal page

Select the Create Reversal Batch check box and specify the period in which you want the reversing journal entry to occur.

You can specify the next period based on your original journal date or by specifying a specific period in time.

Attaching Supporting Documentation for Manual Journal Entries

Access the Journal Attachments page (EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Journal Entries, Journal Attachments).



Journal Attachments page

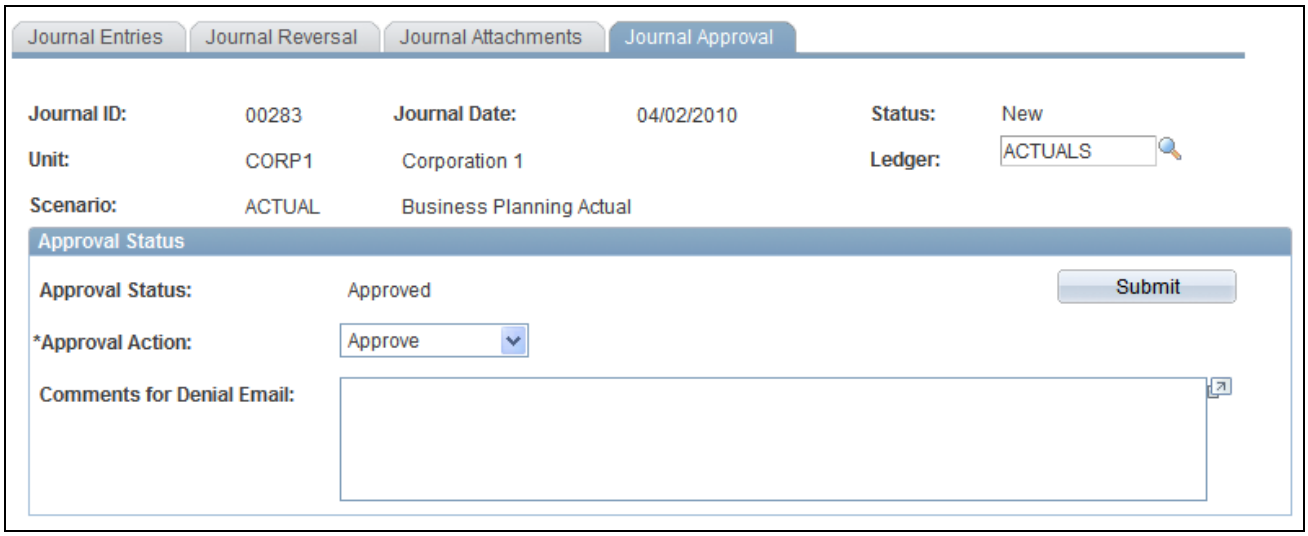
This page enables you to add supporting documentation for your manual journal.

Click the Add Attachment button to locate the desired supporting documentation, select the file, and attach it to the page.

To view a document, click the File Name link. To remove a document, click the Delete button.

Approving Journals

Access the Journal Approval page (EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Journal Entries, Journal Approval).



Journal Approval page

Submit for Approval Click to submit this journal entry for approval.

You set up approval rules on the PF Unit Scenario Definition page.

Approval Action Select an approval status for the journal entry.

You can select from the following values:

None: Indicates a new, unsaved, or saved journal, requiring approval.

Approved: Indicates a journal that is approved (or was preapproved) and is eligible for posting.

The system does not copy journal lines into the journal fact table until the journal status is set to *Approved.* As a result, the journal is ineligible for posting until approved.

Pending: Indicates that the journal is submitted and awaiting approval.

When multiple approval steps are in place, the journal remains in pending approval status until all necessary parties approve the journal.

Denied: Indicates that the journal is denied approval.

Comments for Denial Email Enter any comments that you want to include in the denial email as a result of the journal being rejected.

To update the approval status for a journal, select an approval action from the drop-down list box, and then click the Submit for Approval button.

Processing and Posting Journals

This section provides overviews of journal processing and posting and the PF Journal Modification component and discusses how to:

- Run PF Journal Edit and PF Journal Re-Edit.
- View journal error statistics.
- Correct journal errors.
- Run PF Journal Cleanup.
- Run PF Ledger Post.
- Run PF Ledger Unpost.
- Unpost and repost transactions.

Understanding Journal Processing and Posting

After you set up and run engines such as Data Manager or Ledger Mapper, the enriched data is posted to a temporary table as specified on the Ledger Template page. You then run the PF Journal Edit engine to identify any balance errors and check for valid dimension values in the ChartFields that are specified on the journal entry. Valid data is written to the PF_JRNL_F00, while any errors that are found are written to the error table, PF_JRNL_E00. Use the PF Journal Modification engine to revise errors, and then run PF Journal Re-Edit to repopulate the PF_JRNL_F00 table. Finally, run the PF Ledger Post engine to populate the PF_LEDGER_F00 table with valid and accurate data from PF_JRNL_F00.

The PF Journal Edit process checks for TSE field errors. This process also checks the balancing rules for the jobs in the jobstream to verify whether the flash totals from these jobs are in balance based on the balancing rules that you defined for the jobs.

Input to the PF Journal Edit engine is always from the performance journal temporary table that is populated by other application engines. For this reason, you must always run the PF Journal Edit engine as a job after the application engine that populates the temporary journal table, but within the same jobstream. PF Journal Edit can never be the first or the only job in a jobstream.

Note. The edit process validates against any prompt tables within the record definition.

The purpose of the PF Journal Edit engine is to validate journals and ensure data integrity before you post to PF_LEDGER_F00. After you run PF Journal Edit for any journal, correct any TSE errors using PF Journal Modification, and then run a PF Re-Edit job. If you have missing dimensions in your first edit, use EPM Foundation dimension pages to add the missing dimensions.

The following table lists the PF Journal Edit batch statuses. These statuses appear on the Ledger Post - Batch Selection page. The system creates one record for each batch status change in the PF_Batch table to allow tracking of the batch status history:

Batch Status	*Description*
E	Batch has TSE errors and balance errors.
T	Batch has TSE errors but no balancing errors.

Batch Status	*Description*
B	Batch has no TSE Errors but has balancing errors. A batch status is set to *B* if any of the balancing rules that are defined for the jobs in the jobstream with a balance rule type of control or force balance and the balancing rule total has an amount difference. The amount difference for the balancing rule with a force balance type is written to the PF_JRNL_F00 for the account that is related to the PF ledger event code and is defined for the balancing rule.
V	This is a valid batch. No TSE or balance errors exist. Batch is ready to post.
P	Batch has been posted to the ledger. Status is set by PF_POST engine.
S	Valid data for the batch has been force-posted to the PF Ledger table. Invalid data (TSE errors) still exists in PF_JRNL_E00 that can be modified and posted later. Status is set by PF_POST engine.
F	Batch has been force-posted to the PF Ledger table even though balance errors occurred for the batch. You can address the balancing error by posting the difference to the PF Ledger (force balancing). Status is set by PF_POST engine.
O	Batch has been force-posted to PF Ledger table even though balance and TSE errors occurred. Status is set by PF_POST engine.
U	Batch has been unposted from PF Ledger table. Status is set by PF_UNPOST engine.

Note. If the batch is valid, the data is moved to the PF_JRNL_F00 table. If the batch has TSE errors, you must run the PF Journal Modification engine. After correcting errors in PF Journal Modification, use the Journal Re-Edit process to move the corrected data to the PF_JRNL_F00.

The difference between the Journal Edit and Journal Re-Edit processes is that Journal Edit must always be run as a job after the application engine that populates the temporary journal table, but within the same jobstream. (The input file for Journal Edit engine is temporary journal table). You can run Journal Re-Edit as a standalone job because its input file is a permanent table (PF_JRNL_E00).

Journal Re-Edit performs the following functions for all batches: It checks all the batches in PF_JRNL_E00 that have the same keys as the run parameters (business unit, scenario ID, fiscal year, and period) for TSE errors.

Understanding the PF Journal Modification Component

Use the PF Journal Modification component to correct any data that was flagged as an error during the PF Journal Edit process and moved to the PF_JRNL_E00 error table. After correcting errors, run the Journal Re-Edit process to perform TSE validation on data in the PF_JRNL_E00 table and change the data error flag to *No* if the data is no longer erroneous. (The Journal Post process moves corrected data in PF_JRNL_E00 to PF_JRNL_F00).

The PF Journal Modification pages are designed to enable you to correct actual dimension *values* in the data, not the dimension tables. You must use EPM Foundation Dimension pages to correct the dimension tables.

Journal Re-Edit performs TSE validation against all the error records in PF_JRNL_E00 that have the *same* run control parameters (business unit, fiscal year, accounting period, scenario ID). This means that if you run multiple edits for the same parameters at different times, (thus creating multiple batches with the same runtime parameters), you can correct all the batches and dimensions first, and then run Journal Re-Edit once for all batches. Likewise, you can correct one batch at a time, and run Journal Re-Edit after correcting each batch.

Pages Used to Process and Post Journals

Page Name	*Definition Name*	*Navigation*	*Usage*
PF Journal Modification - Journal Statistic	PF_JRNL_STATS	EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Correct Errors, PF Journal Modification, Journal Statistic	View journal error statistics by viewing batches of data that are flagged as errors and moved to the PF_JRNL_E00 error table.
PF Journal Modification - Journal Correction	PF_JRNL_CORRECTION	EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Correct Errors, PF Journal Modification, Journal Correction	View error fields and the number of errors.
PF Journal Modification - Error Details	PF_JRNL_SUMM	Click the Display Record Errors button on the Journal Correction page.	Correct errors as appropriate.
PF Journal Modification - Error Description	PF_JRNL_TSE	Click the Display Error Messages button on the Error Details page.	View error message details for an error.
Journal Cleanup	RUN_PF_DELB	EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Journal Cleanup Report, Journal Cleanup	Run PF Journal Cleanup to delete journal data by batch ID.

Page Name	Definition Name	Navigation	Usage
Journal Cleanup - Batch Selection	PF_DELETE_BATCH	Click the Get Batch Data link on the Journal Clean Up page.	Specify whether to delete all displayed batches or use the check box to delete one batch at a time.
Ledger Post	RUN_PF_POST	EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Post Ledger, Ledger Post	Run the PF Post engine to post data in the journal table that you have determined is valid.
Ledger Post - Batch Selection	PF_POST_BATCH	• EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Post Ledger, Batch Selection • Click the Get Batch Data link on the Ledger Post page.	Specify the batches that you want to post to the performance ledger.
PF Ledger Unpost	RUN_PF_UNP	EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Unpost Ledger, PF Ledger Unpost	Run the PF Unpost to unpost any posted data by batch ID that you determine is invalid.
PF Ledger Unpost - Batch Selection	PF_UNPOST_BATCH	• EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Unpost Ledger, Batch Selection. • Click the Get Batch Data link on the Ledger Unpost page.	Specify the batches that you want to unpost from the performance ledger.

Running PF Journal Edit and PF Journal Re-Edit

The PF Journal Edit process checks for TSE field errors. If the edit process does not find any errors, the data is moved to the PF_JRNL_F00 table. If the edit process does find errors, you must correct them before proceeding. After correcting the errors you must run the Journal Re-Edit process which performs TSE validation on data in the PF_JRNL_E00 table, changes the data error flag to *No* if the data is no longer erroneous, and moves corrected data in PF_JRNL_E00 to PF_JRNL_F00.

Note. In EPM, you can use the performance ledger and the Average Daily Balance (ADB) ledger. Use of the ADB ledger is optional and all of the processing in the ADB ledger is the same as in the performance ledger. The type of balance that is stored is the only difference between the two ledger tables.

PF Journal Edit

You must run PF Journal Edit as a job after running the engine that loads journal data to the performance journal temporary table.

PF Journal Re-Edit

To run PF Journal Re-Edit, follow these steps:

1. Add the jobstream *RE_EDIT* to the *Jobstream* page and select *RE_EDIT* for the job ID.

   Note. You must run PF Journal Re-Edit as a stand-alone job in its own jobstream.

2. Using the *Jobstream Record Suites* page, add the appropriate record suites to the *RE_EDIT* jobstream.

3. Using the *Journal Re-Edit Report* page, create the run control for the *RE_EDIT* jobstream.

   The *Journal Re-Edit Report* page can be accessed using the following navigation: EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals.

   Warning! You should never run this jobstream from the *Run Jobstream* page! You cannot specify a ledger ID on the *Run Jobstream* page and the engine will fail without it.

When defining your run control parameters, remember that the difference between the Rerun check box and the PF Journal Re-Edit engine is that the PF Journal Re-Edit engine is used to re-edit previous batches of journal entries that had TSE errors. Rerun is used after another *batch with the same parameters* has already been run.

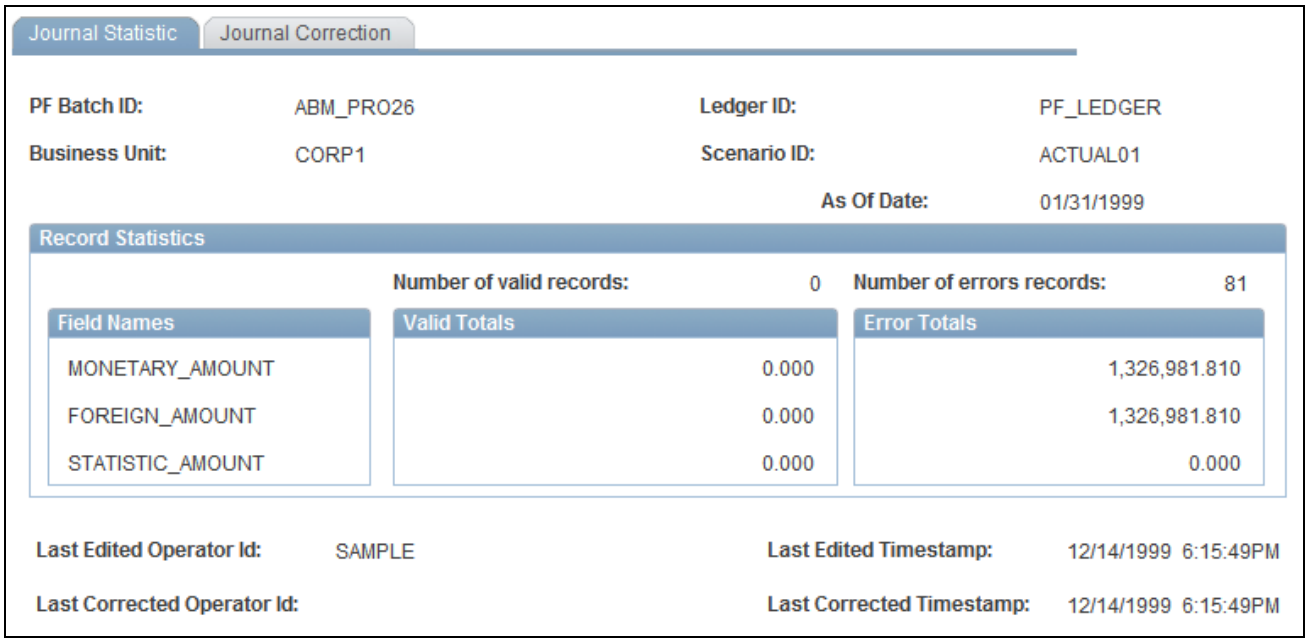
If the status of the batch is other than posted (P,S, F, O) or unposted (U), you can rerun the batch by selecting the Rerun check box. The batch ID is added to the PF_DELBATCH_TBL and is used to clean up the data in the journal for the batch using the PF Delete Batch utility.

See Also

Chapter 20, "Streamlining Processing with Jobstreams," page 461

Viewing Journal Error Statistics

Access the Journal Statistic page (EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Correct Errors, PF Journal Modification, Journal Statistic).



Journal Statistic page

PF Batch ID The PF Journal Edit process assigns a PF batch ID to a batch of data going into the journal table. It uses the PF_BATCH_NUMBER that is stored in the warehouse business unit table as the next batch number. The edit process increments this batch number by one every time it processes a new batch of data. The PF batch ID is a combination of the scenario ID, the fiscal year, and the accounting period.

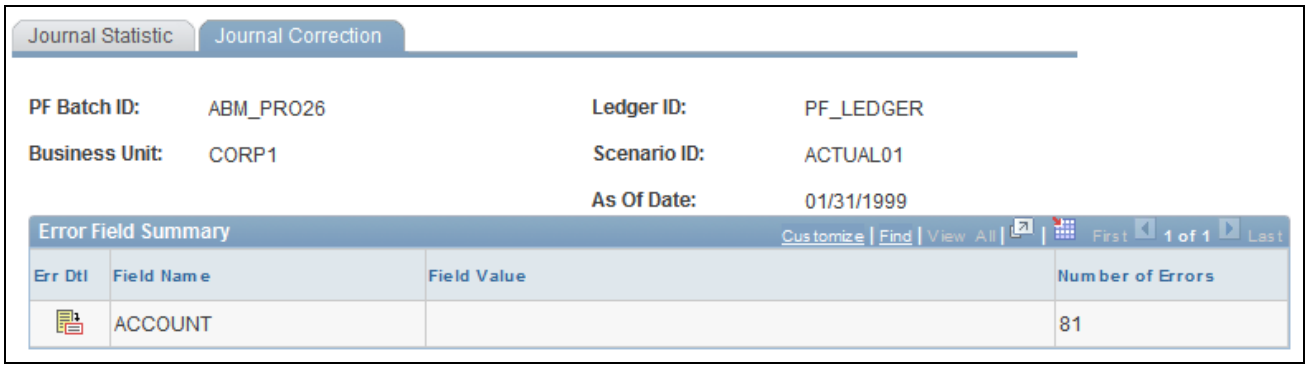
Business Unit, Ledger ID, Scenario ID, and **As Of Date** These parameters were specified in the run control that was used to run the jobstream that contained the PF Journal Edit engine. The business unit and PF batch ID uniquely identify a batch of data to be modified.

Record Statistics Use the Number of valid records and Number of error records display fields for an overview perspective on the amount of errors.

Last Edited Operator ID, Last Corrected Operator ID, Last Edited Timestamp, and **Last Corrected Timestamp** These fields display information about who ran the last journal edit and when it was run.

Correcting Journal Errors

Access the Journal Correction page (EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Correct Errors, PF Journal Modification, Journal Correction).



Journal Correction page

Field Name Displays the name of the field in error.

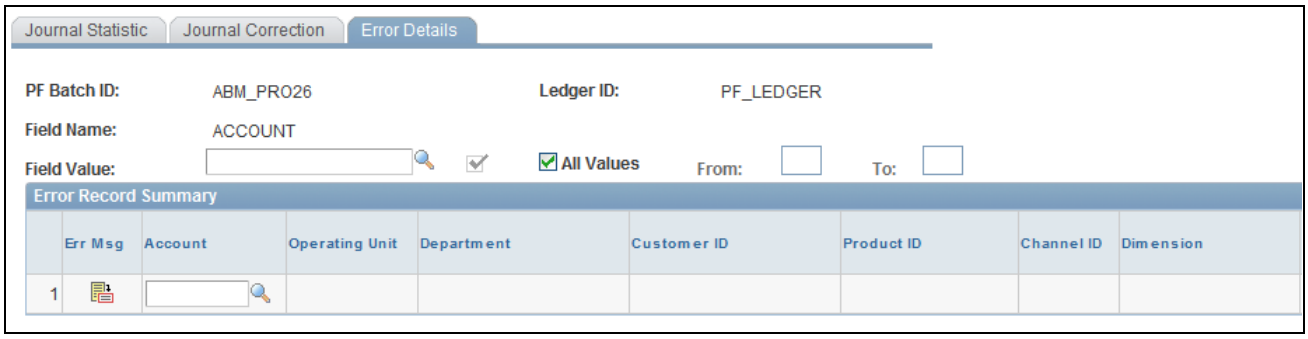
Field Value Displays the value for the field in error.

Number of Errors Displays the total number of errors.

Click the Error Details button to access the Error Details page, where you can view the error details for the field and values in error. You can also correct the dimension value to correct the error.

Correcting Errors

Access the Error Details page (Click the Error Details button on the Journal Correction page).



Error Details page

Field Value Specify a field value (dimension) to apply to the lines in error.

All Values Select this check box to apply this value to all the rows that are listed. Deselect this check box to apply the value to specific rows only. Specify the rows using the From and To fields.

From and **To** Specify the row numbers to which the new field value should be applied.

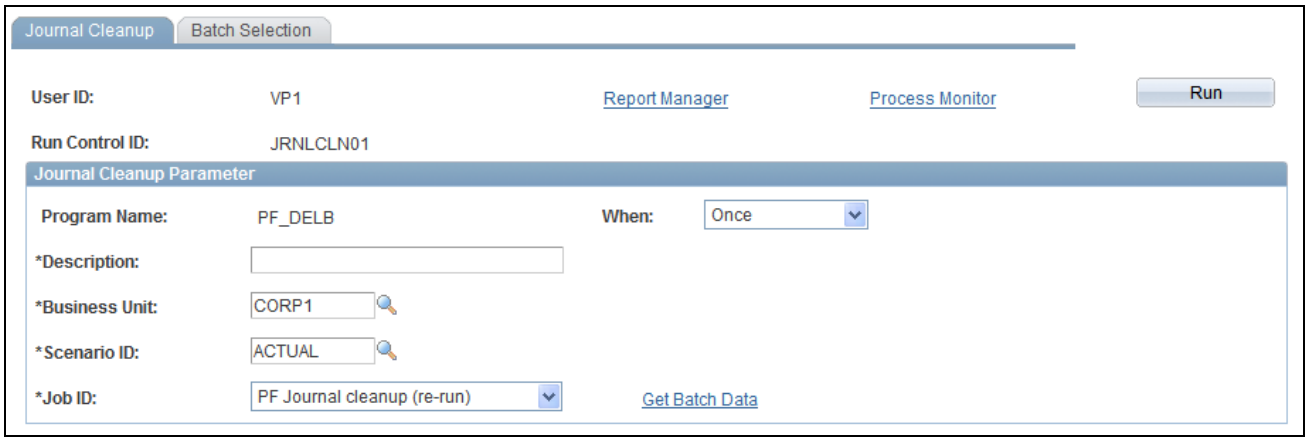
Specify a dimension to apply to the values in error.

	Click the Display Error Messages button on this page to access the Error Description page for that error and review the error message.

Save your changes.

Running PF Journal Cleanup

Access the Journal Cleanup page (EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Journal Cleanup Report, Journal Cleanup).



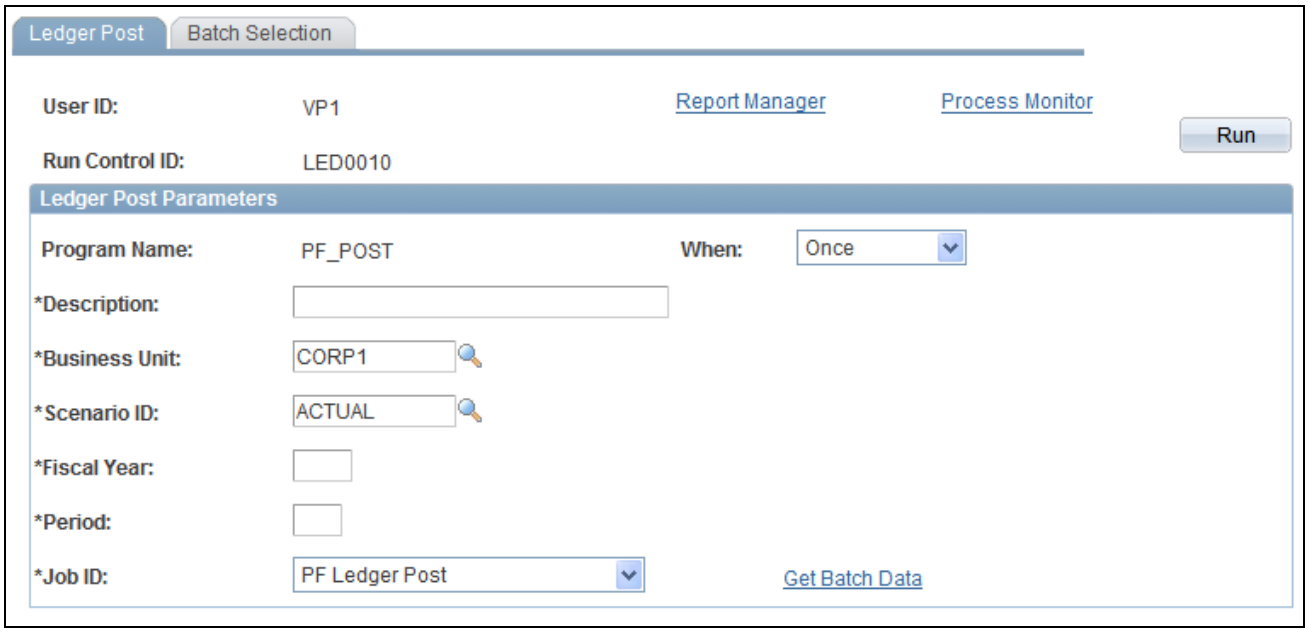
Journal Cleanup page

Business Unit and **Scenario ID**	Select the business unit and scenario ID for which to delete batches.
Job ID	Select *PF Journal cleanup* for this standalone job.
Get Batch Data	Click to retrieve all batch data for the specified business unit and scenario ID and access the Batch Selection page.

On the Journal Cleanup - Batch Selection page, select the batches to clean up by selecting the Delete Batch Flag check box. Save your changes and return to the Journal Cleanup page to run the PF_DELB process.

Running PF Ledger Post

Access the Ledger Post page (EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Post Ledger, Ledger Post).



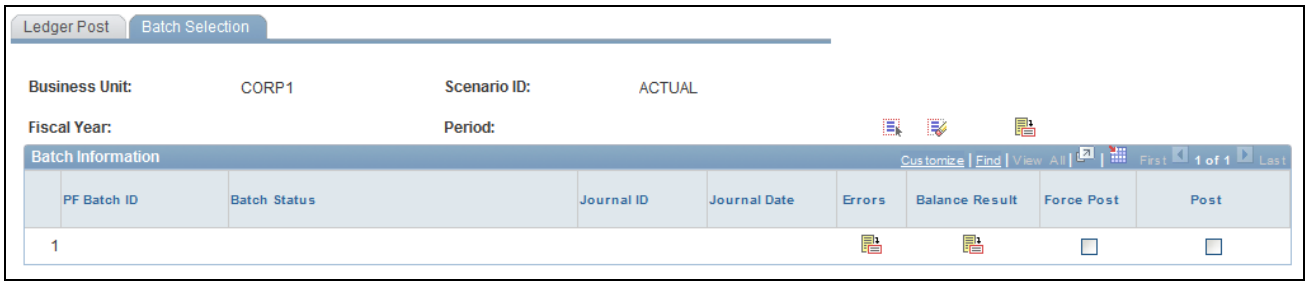
Ledger Post page

Business Unit, Scenario ID, Fiscal Year, and **Period**	Select the business unit , scenario ID, fiscal year, and accounting period for which to post batches.
Job ID	Select *PF Ledger Post* for this standalone job.
Get Batch Data	Click to access the Batch Selection page and retrieve all batch data for the specified parameters.

Note. PF Ledger Post is the last step in the data movement and migration process before you generate reports. The data that reaches the performance ledger table (PF_LEDGER_F00) must be accurate. You can post multiple batches at one time. The only way to post to the ledger is from the performance journal table (PF_JRNL_F00).

Batch Selection

Access the Batch Selection page (Click Get Batch Data on the Ledger Post page).



Batch Selection page

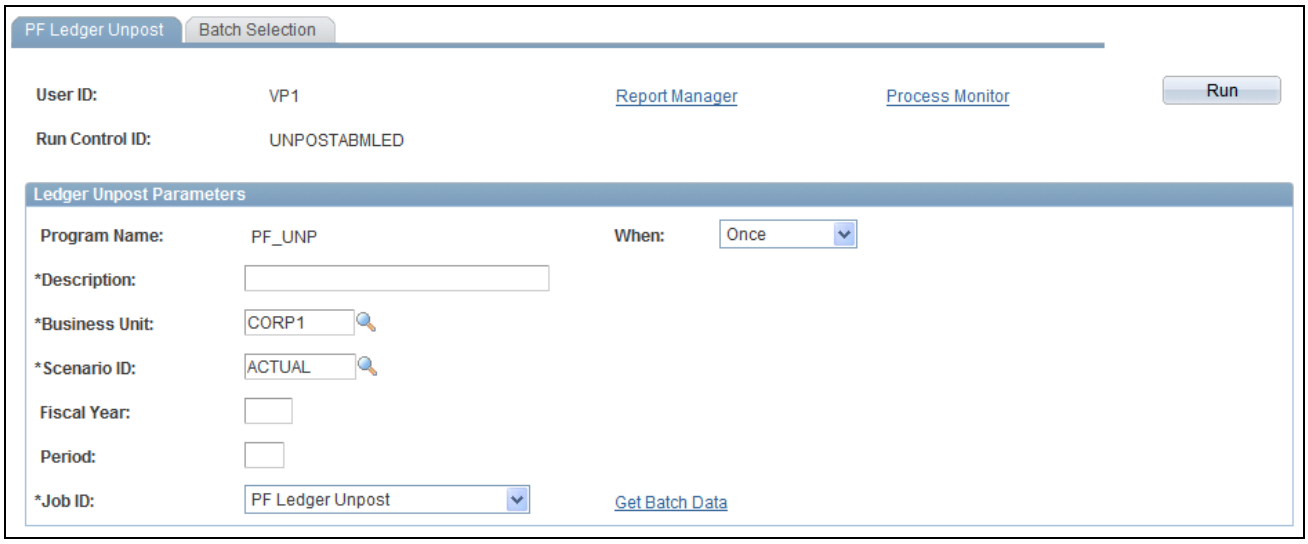
	Click the Get Batch Data button to retrieve the batch data. The data appears in the Batch Information grid.
	Click the Display Journal Error Report button to access the PF Journal Modification component to view and correct any errors. If you choose not to correct the errors, select the Force Post check box to post this batch to the performance ledger table.
	Click the Display Balance Results button to see any balance errors that might exist. The display-only Balance Error Statistics page appears. If you choose not to correct the balance errors, select the Force Post check box to post this batch to the performance ledger table.
Force Post	Select this check box to force post batches that are in error to the performance ledger table. If a balance rule with force balance is used, the out-of-balance amount is posted to the account that is related to the ledger event code for the force balance rule. Only one force balance rule is allowed per batch.
Post	If you do not have any errors for the batches, the Post check box is automatically selected.

Save your changes and return to the Ledger Post page to run the PF Ledger Post engine and post the specified batches to the performance ledger table.

If you believe that you have posted in error, run the PF Unpost engine.

Running PF Ledger Unpost

Access the PF Ledger Unpost page (EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Unpost Ledger, PF Ledger Unpost).



PF Ledger Unpost page

Business Unit, Scenario ID, Fiscal Year, and **Period** — Select the business unit, scenario ID, fiscal year, and accounting period for which to unpost batches.

Job ID — Select *PF Ledger Unpost* for this standalone job.

Get Batch Data — Click to access the Batch Selection page and retrieve all batch data for the specified parameters.

Note. The functionality of the Batch Selection page is identical to Batch Selection page for Ledger Post; please refer to that documentation for reference.

Note. Running the PF Ledger Unpost engine does not necessarily zero-out all the posted data. If the keys are the same for PF batch data during the PF Post process, the data is aggregated in the performance ledger table for those batches. If you unpost aggregated data, you might not obtain a zero balance unless you unpost all the batches for the same parameters. Use the run control ID to find the batches that you want to unpost.

Unposting and Reposting Transactions

To ensure data integrity, PeopleSoft designed the system to prevent you from reposting a batch that has already been posted. Suppose that you post a batch of transactions called Batch A and subsequently find that the data is invalid. You unpost the batch and make the necessary changes. To repost the batch, you must complete the following steps:

To repost a batch previously posted:

1. Run the entire jobstream again to repopulate PF_JRNL_F00 and create a new batch ID (in our example Batch B).

   Your jobstream might include running engines such as Activity-Based Management, File Transfer Protocol, Data Manager, Allocation Manager, or Ledger Mapper, and the PF Edit engine. On the PF Journal Edit run control, make sure that you select the Rerun check box to ensure that the system flags the previously posted batch (Batch A in our example) to be deleted.

2. Use PF Journal Cleanup to delete obsolete records in PF_JRNL_F00 from the previously posted batch (Batch A).

3. Post the transactions to PF_LEDGER_F00 using the new PF batch ID (Batch B).

Note. You can also copy the journal to a new journal and post the copy.

Using Balancing and Reconciliation Features

This section provides overviews of PF Reconciliation, job total metadata, and balance rules metadata and discusses how to:

- Review and define job totals metadata.
- Review and define balance rules metadata.
- Run PF Reconciliation.
- Review reconciliation results.

Understanding PF Reconciliation

Several features are available for you to use to check data integrity and to verify that input equals output between tables.

Run the PF Reconciliation engine after populating the performance ledger table (PF_LEDGER_F00) to verify that the data that you processed through the system is accurately reflected. The job totals and balance rules metadata pages enable you to specify the fields in the tables that you want to track. You define and use job totals metadata to compare totals between sources such as the GL ledger table or the revenue table versus the target table and the performance ledger table, and find any differences for an accounting period of a business unit and scenario ID.

The PF Reconciliation engine first resolves the sources and destinations into temporary tables using the constraints that are given for the job totals. The engine verifies all the job totals and balancing rules that are defined under the reconciliation job that is being run and calculates the totals and the differences for the balancing rules.

Typically, you run the PF Reconciliation engine after a period to compare to and from amounts between tables, such as the REVENUE_F00 and the PF_LEDGER_F00 or the GL_LEDGER and the PF_LEDGER_F00. The system delivers job totals and balance rules metadata for these types of tables. However, if you want to compare any other source and target tables, you can create your own job totals and balance rule metadata. After running your reconciliation, you can generate a Crystal report for review. You specify the Crystal report ID on the Balance Rules Metadata page. Additionally, use the inquiry pages to track your reconciliation job.

Understanding Job Total Metadata

Job total metadata has two purposes:

1. To define totals for the PF Record Summary.

   For PF Record Summary, define job totals using the Record Name field.

2. To define totals for balancing rules.

   For balancing rules, you define job totals using constraints. Constraints are used to define the scope of the data that is being totaled. The data could come from a single table, but the scope of the data to be considered could depend on data in other tables.

Note. You can toggle the record and constraint fields on the Job Totals page, depending on the job ID that you select. Predefined job total metadata is delivered with EPM .

Understanding Balance Rules Metadata

Balance rules enable you to verify the data throughout the system. For example, you can verify that a monetary amount that you entered into the system reached the performance ledger table (PF_LEDGER_F00). The three balancing rules include:

Informational	Use this balancing rule to review how many rows or how many monetary amounts are in the specified table, or to see information that is not critical but can be used to validate or identify a model. For example, for PeopleSoft Activity-Based Management, you can check that data reaches cost objects directly from resources instead of going through activities. Informational balance rule differences enable you to post to the ledger.
Control	Use this balancing rule to determine whether an out-of-balance amount exists. The batch is set to balance error status and can be analyzed in a report. Control balance rule differences will not allow you to post to the ledger.
Force Balancing	An out-of-balance amount will not stop the post process and can be added to the performance ledger table to force balance. Force balancing can be defined on a rule that compares totals between a source and a destination. You can define only one balance rule for a source and a destination. The setID MODEL has a predefined force balance account (FBAL). If you need to define a force balance account, do so on the Ledger Event Codes page.

Predefined balance rules metadata is delivered with EPM.

See Also

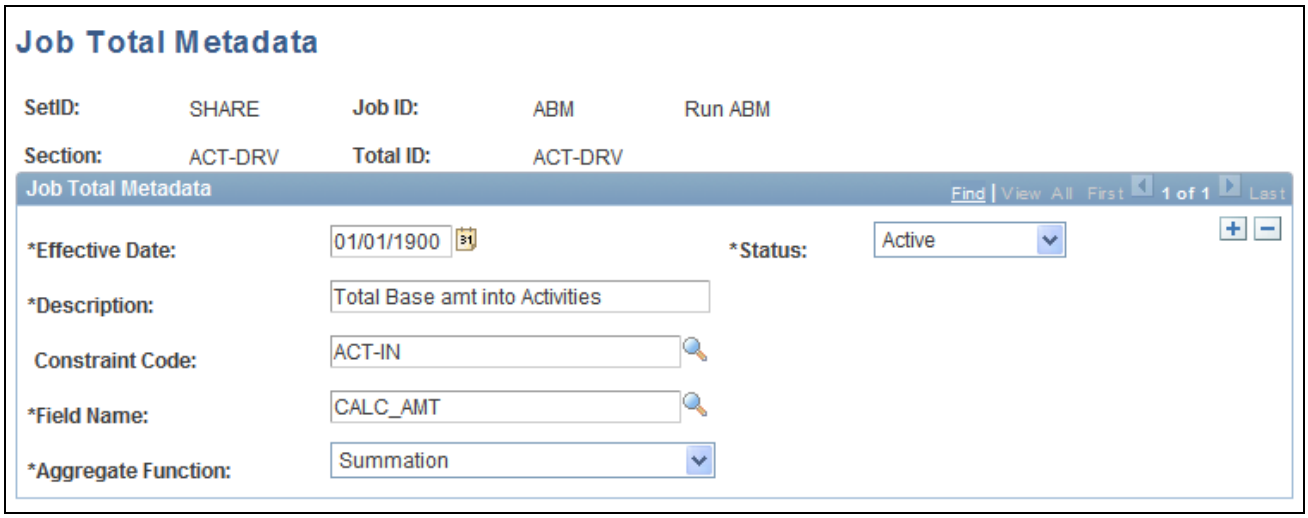
Chapter 18, "Setting Up Business Rules for the Operational Warehouse - Enriched," Defining Ledger Event Codes, page 429

Pages Used to Set Up Balancing and Reconciliation

Page Name	*Definition Name*	*Navigation*	*Usage*
Job Total Metadata	PF_META_TOT_TBL1	EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Job Total Metadata	Review and define job totals for various data that is used as input or output to the different source or target tables in the system.
Balance Rules Metadata	PF_META_BAL_TBL1	EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Balance Rules Metadata	Review and define balance rules.
PF Reconciliation	RUN_PF_JOB	EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Reconcile Balances, PF Reconciliation	Run the PF Reconciliation engine to compare to and from amounts between tables.
Job Totals	PF_RECON_TOT_DATA	EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Review Bal/Recon (Balance/Reconcile) Results, Job Totals	A display-only page that shows the results of a job totals reconciliation.
Balance Rule Data	PF_RECON_RULE_DATA	EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Review Bal/Recon (Balance/Reconcile) Results, Balance Rule Data	A display-only page that shows the results of a balance reconciliation.

Reviewing and Defining Job Totals Metadata

Access the Job Total Metadata page (EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Job Total Metadata).



Job Total Metadata page

Description The description that you enter here is used by the Metadata Search engine to find your metadata.

Constraint Code Select a constraint code. Constraints are used in this case to define the scope of the data that is being totaled. This data could come from a single table, but the scope of the data to be considered could depend on data in other tables.

This field is not available if you are setting up job totals for the SUMM Job ID (PF Record Summary engine). For PF Record Summary, job totals are defined using the Record field.

Field Name Select a field name. This field displays the primary record for the field name that you select. For record summary totals, select the record name. The field name identifies the field that you want to total. If the list box does not list the field that you expected, check your datamap definition.

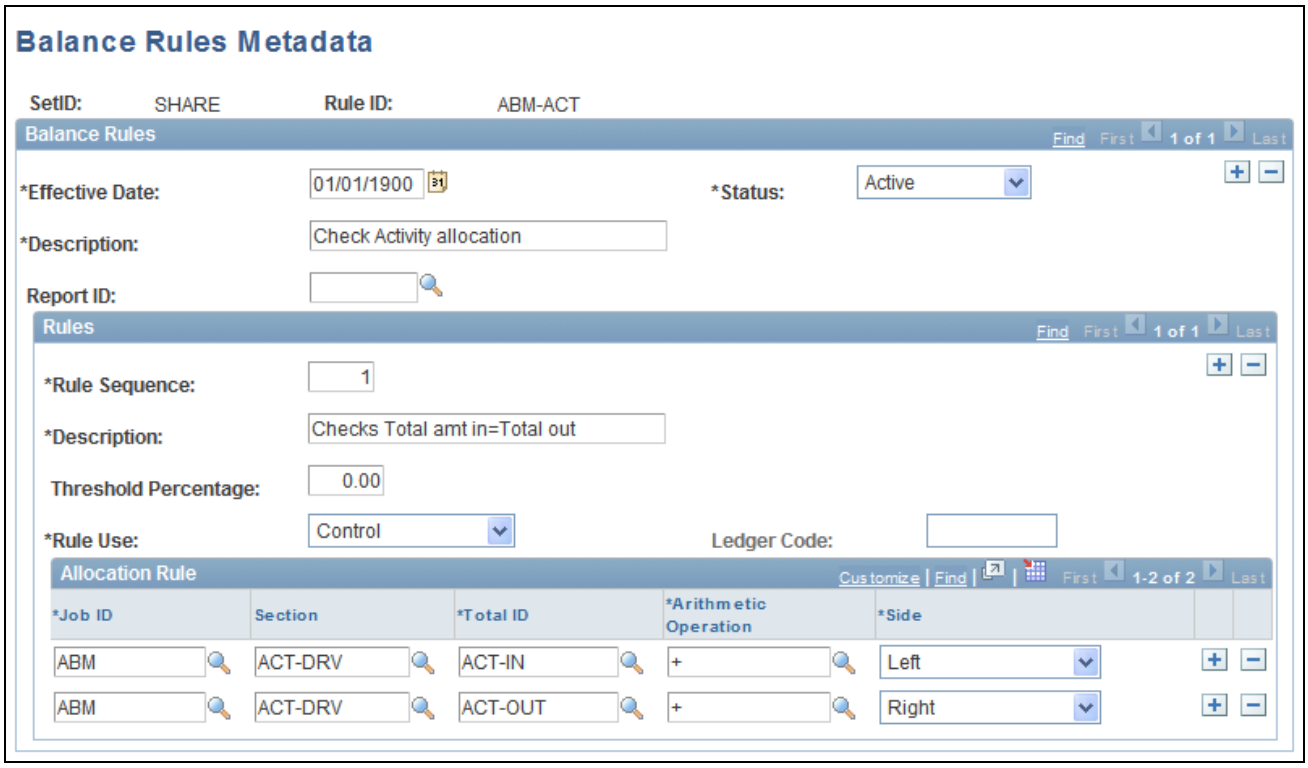
Aggregate Function For aggregate functions, you usually select *Summation* to ensure that all the amounts in the specified field are summed.

Select *Count* to count total records in the data (for example, to count the number of employee IDs).

Select *Average* to average the specified field name totals.

Reviewing and Defining Balance Rules Metadata

Access the Balance Rules Metadata page (EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Balance Rules Metadata).



Balance Rules Metadata page

Description The description that you enter is used by the Metadata Search engine to find your metadata.

Report ID Select the ID of the report that you want to generate to help analyze any problems that caused the balance differences. You can use this ID for generating a Crystal report after you run the PF Reconciliation engine.

Rule Sequence Enables you to group similar rules as subrules. To compare more than one balance field amount, click the add button in the Rule Sequence field to add a rule for a different balance amount field. For example, to compare posted total amounts, create rule sequence 1 and then click the Add button to add rule sequence 2 to compare base amounts.

Description Enter a description that accurately defines the balance rule.

Threshold Percentage Specify a percentage that reflects the highest percentage that you want to use for analysis. For instance, entering *5.00* in this field would indicate that if the difference between resources and activities in Activity-Based management is over five percent, the system should display the amounts. If the difference is not over five percent, post to the performance ledger table.

Rule Use Select from the following rule uses:

Rule Use	*Description*
Control	Select *Control* if you are comparing tables for engine output amounts. For example, you would select *Control* and enter a threshold percentage if you were tracking whether the amount for resources equaled the amount that was driven to activities in Activity-Based Management. Any balance error is assigned a batch error status. Generate reports to analyze the balance error before posting.
Force Balance	Use *Force Balance* to post balance rule differences that should be posted to the PF Journal table. Then select the ledger event code that identifies the account to which the differences will be posted in the PF Ledger table.
Info Only (information only)	Select *Info Only* if you are comparing totals that are not critical (for example, the number of records). *Info Only* does not create a balance error batch. The batch will be valid.

Ledger Code For a rule use of *Force Balance*, select the account to which the differences that are generated by force balancing should be posted in the performance ledger table.

Job ID Select the job ID that is defined on the Job Metadata page.

Section This is not a required field. Use this field to help you identify the problem area when an-out-of balance error occurs. The system uses the prompt table PS_PF_META_SEC_VW3. Before a section can be considered valid, it must first be defined in the engine metadata. Valid section codes that you enter on the engine metadata page are the actual application engine sections within the application engine program.

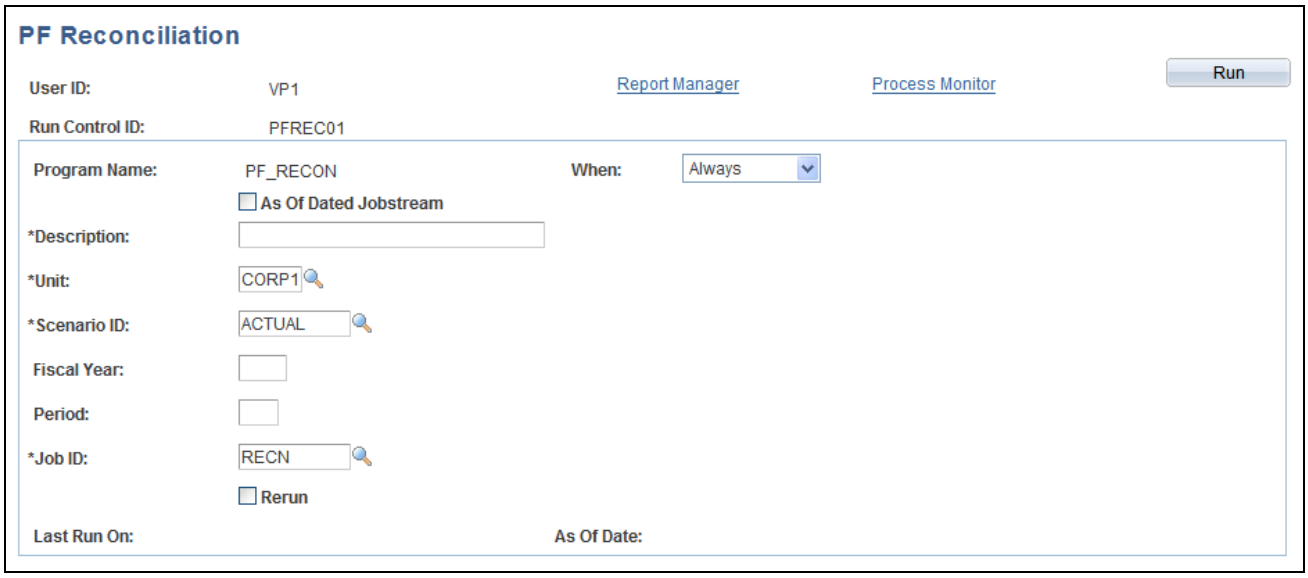
Total ID Select the total ID that is defined in the Job Totals Metadata page.

Arithmetic Operation Select an arithmetic operation (for example, + or −) to add or subtract multiple job totals. These operations are used with the Side field

Side Select *Left* for the input total IDs, and *Right* for the output total IDs.

Running PF Reconciliation

Access the PF Reconciliation page (EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Reconcile Balances, PF Reconciliation).



PF Reconciliation page

The parameters on this page are described in the chapter Streamlining Processing with Jobstreams earlier in this PeopleBook.

After running PF Reconciliation, you can view your balance and reconciliation results using the Review Bal/Recon Results (review balance reconciliation results) component.

Note. You must include the PF Merge engine in a jobstream with the PF Reconciliation engine.

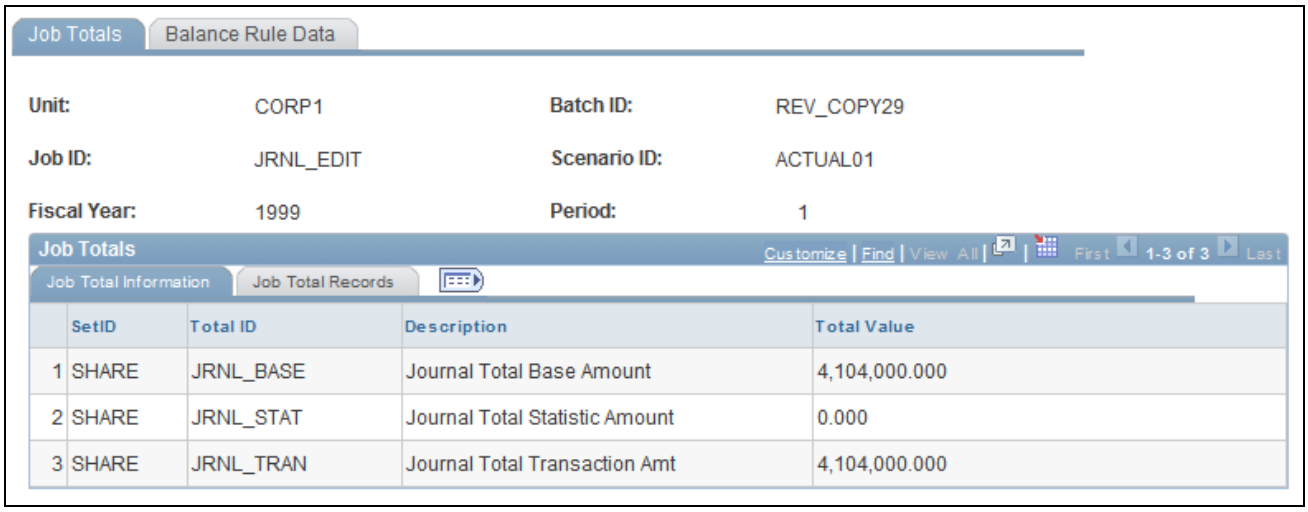
See Also

Chapter 20, "Streamlining Processing with Jobstreams," Running Jobstreams, page 485

Reviewing Reconciliation Results

Access the Job Totals (EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Review Bal/Recon (Balance/Reconcile) Results, Job Totals) and Balance Rule Data pages (EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Review Bal/Recon (Balance/Reconcile) Results, Balance Rule Data).

Use the Job Totals page to review the job total information from your job. For example, you could use the total value on this page to verify that for Activity-Based Management, the total amount that was allocated to resources was transferred to the CALC_OBJ_F00 table.



Job Totals page

This is a display-only page that shows the results of a balance reconciliation. If the Release Record Suite check box is selected, then this job has been released from a record suite.

Using Ledger Drill Down

After loading the performance ledger table (PF_LEDGER_F00), you can track data movement to and from the PF_LEDGER_F00 using the drill down feature. You can track performance ledger details and general ledger mapper details.

To build a search for ledger drill down, you must include the business unit, fiscal year, and accounting period when you specify your search keys.

This section discusses how to:

- Specify drill-down criteria.
- View drill-down details.

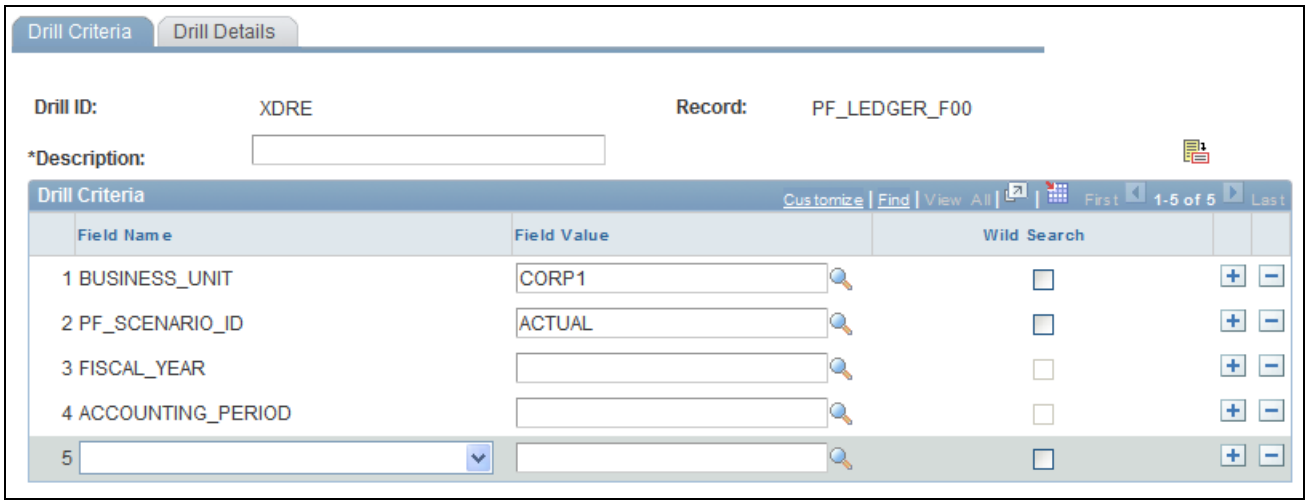
Pages Used to Drill Down on Ledger Data

Page Name	*Definition Name*	*Navigation*	*Usage*
Drill Criteria	PF_DRILLSRCH_TBL1	EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Ledger Details, Drill Criteria	Specify drill criteria for tracking data movement to and from PF_LEDGER_F00.

Page Name	Definition Name	Navigation	Usage
Drill Details	PF_DRILL_VW2	EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Ledger Details, Drill Details	Review and drill down to the results in the PF_LEDGER_F00 table.

Specifying Drill Criteria

Access the Drill Criteria page (EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Ledger Details, Drill Criteria).



Drill Criteria page

Drill Criteria

Specify the drill criteria on which you want to perform the search. You can add further search criteria by adding a row and selecting the field on which to search. At a minimum, you must specify the business unit, scenario ID, fiscal year, and accounting period.

BUSINESS_UNIT Select the business unit.

PF_SCENARIO_ID Select the scenario ID.

FISCAL_YEAR Specify the fiscal year.

ACCOUNTING_PERIOD Specify the accounting period.

Field Value Specify your selections for any additional criteria that you add. The drill-down functions in the order in which you add the rows.

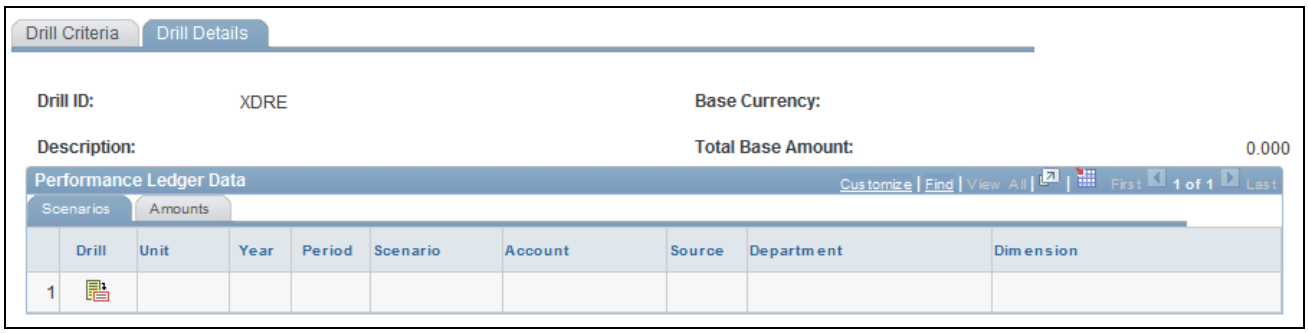
Wild Search Alternately, you can select this check box to specify any values on which you want to search for the specified field.

When you have entered your criteria, click the Get Drill Data button to initiate your search.

The system retrieves the performance ledger data based on the entered search criteria. You can view the details on the Drill Details page.

Viewing Drill Details

Access the Drill Details page (EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Ledger Details, Drill Details).



Drill Details page

Use this page to view the performance ledger data that is based on your search criteria. You can view scenarios and amounts.

Click the Get More Drill Back Data button to access a detail page for a row of data. For example, if you are viewing data in Activity-Based Management, clicking the Get More Drill Back Data button accesses the ABM Ledger Data page (AB_DRILL_TBL2).

Reviewing Profit Manager Error Messages

When running any of the engines that are described in this chapter, you can use the Engine Messages component, which is accessible from the Performance Ledger, Performance Journals menu, to review the progress of the engine and errors that were generated.

This component is described in detail in the chapter on Streamlining Processing with Jobstreams.

See Also

Chapter 20, "Streamlining Processing with Jobstreams," Viewing Engine Messages, page 494

Correcting Profit Manager Fact Table Errors Using PF Modification

This section provides an overview of fact table error correction and discusses how to:

- View PF summary statistics.
- View and correct errors.

Understanding Profit Manager Fact Table Error Correction

After you have run the PF Summary process on a Profit Manager fact table, you can view and correct any errors that were found by opening the appropriate Profit Manager Fact Correction page. This process enables you to view and correct fact errors for your Profit Manger fact tables. For instance, you can use the Ledger Correction component to review and correct invalid data in the ledger fact table. Although there is a separate menu item and page for the Profit Manager Fact Correction process, the process is exactly the same for each.

There is an error table for each Profit Manager fact table. The error table contains all rows with errors that were found during data migration and is a mirror image of the target table, plus two extra columns called Error Flag and PF_EDIT_SEQ_NUM.

Note. Journal modifications are run separately using PF Journal Modification.

For each modification that you run, you must verify the delivered record metadata, record summary metadata, and job totals metadata are accurate.

Viewing PF Summary Statistics

Access the PF Summary Statistics page.

Business Unit and **PF Batch ID** — Uniquely identifies a batch of data to be summarized or modified. The PF Batch ID is a combination of model ID, the fiscal year, and the accounting period.

Number of Valid Records and **Number of Records with Errors** — Review these display fields for an overview of the amount of errors. The field names on the left are defined in job totals metadata and are specified in record summary metadata.

Last Edited Time Stamp — These fields display information about who ran the last PF Record Summary and when the PF Record Summary was run.

Click on the Error Fields tab to continue with the modification process.

Viewing and Correcting Errors

Access the Error Fields page.

Field Name Displays the field name of the error.

Field Value Displays the value for the field name in error.

Number of Errors Displays the number of records in error for the specified field.

You can correct fields by selecting the appropriate value to correct the specified column errors and clicking the Apply button. For other fields, you can click on the individual rows and correct those errors using the prompt table.

Chapter 22

Using Data Enrichment Tools

This chapter provides an overview of the Data Manager and Allocation Manager data enrichment tools and discusses how to:

- Set up value objects.
- Create indices on Allocation Manager temporary tables
- Define Data Manager rules.
- Define Data Manager rule sets.
- Rename Data Manager rules.
- Associate Data Manger rules sets with a job.
- Run the Data Manager engine.
- Define Allocation Manager rules.
- Define Allocation Manager rule sets.
- Rename Allocation Manager rules.
- Associate Allocation Manager rules sets with a job.
- Process allocations.
- Query allocations.

Understanding Data Manager and Allocation Manager

This section discusses:

- Data Manager.
- Allocation Manager.
- Mathematical model for Data Manager and Allocation Manager rules.

Data Manager

The Data Manager is a flexible tool that enables you to:

- Move data into the performance journal table (PF_JRNL_F00).
- Organize the output from any PeopleSoft Enterprise Performance Management engine to create rows of data with multiple dimensions.
- Create reporting categories (dimensions) that represent groups of information for reports, such as business units, customers, products, channels, accounts (revenue, expense, and so on), and departments.

Note these key features of Data Manager:

- You can use Data Manager to consolidate or aggregate data and eliminate redundant rows.
- You can run Data Manager using different rules to organize and create multiple dimensions for direct costs and revenue data.

Data Manager supports the following methods for moving data:

Method	***Description***
Copy	Moves data from the source to the target. This method typically uses multiple-dimension data as the source.
Arithmetic	Applies an arithmetic operation using the amounts in both sources.
GL Mapper	Takes data from the general ledger tables and maps it to performance data. This method populates the temporary GLSTG table . From here, you can use the copy method to move data directly to the performance journal.
Prorata	Adds an additional dimension to your data and divides the amount across the dimension based on the percentage of the total amount.
Spread Even	Adds an additional dimension to your data and spreads the fact amount evenly across the dimension.
Tree Aggregation	Aggregates measures based on a tree hierarchy. The node names act as dimensions.

Data Manager and the Profit Manager

Data Manager works with the Profit Manager tools by moving data to the performance journal table (PF_JRNL_F00). The Profit Manager tools verify this data (PF_EDIT engine) and post the data to the performance ledger (PF_POST engine). They also unpost data, if necessary (PF_UNPOST engine) and can clean up the performance journal table (PF Journal Cleanup engine). The Profit Manager includes tools for balancing and reconciling your data.

The performance ledger, PF_LEDGER_F00, is a useful table for reporting because it contains all of the performance fact data. The facts have been allocated across several dimensions of your choice (typically channel, product, customer, and department). Having all of this information organized in a single table makes it easier for you to query and create reports.

The Profit Manager tool is described in another chapter of this PeopleBook.

See Chapter 21, "Setting Up and Using Profit Manager," page 497.

Data Manager Setup

To set up the Data Manager, you need to:

1. Define the necessary metadata (including tablemaps, datamaps, constraints, and filters).

   The system uses datamaps and constraints to point to the appropriate tables.

2. Define Data Manager rules, including the method by which you want to move, aggregate, or create multidimensional data, as well as the sources and target of the process.

3. Create a Data Manager rule set that contains one or more rules.

4. Set up job metadata and jobstreams.

5. Associate the Data Manager rule set with a job.

6. Run the Data Manager engine.

Allocation Manager

Allocation Manager is an EPM tool that enables you to distribute revenue, expense, and statistical amounts across business units, departments, and other dimensions. For example, you can allocate budget planning to detail levels to perform detailed budgeting.

The Allocation Manager tool can also be used to create offset and residual rows, capture source, basis, and target data, and create, process, and post journals to a ledger.

Allocation Manager consists of several PIA components and an application engine. You create the rules and rule sets that define your processing with PIA pages, while the application engine is executed using a jobstream and a run control PIA page.

Each allocation output is determined by the type of allocation method that you select. The following table lists the types of allocations that are supported by the Allocation Manager and describes each allocation type:

Allocation Type	*Description*
Arithmetic Operation	Defines a mathematical calculation using the source and basis, such as source + basis.
Prorata	Divides the source amount proportionately among the targets based on basis measures.
Spread Even	Distributes the source amounts equally by the specified basis fields. For instance, if expenses were spread evenly across four business units, each would have 25 percent of the expense.

Allocation Type	*Description*
Copy	Copies the source amounts to the targets.

Allocation Manager can use any source dimension within the OWE and provide output to any target. As a result, you can use sources from Global Consolidations and ABM results to further manipulate the output for analysis based on business rules specific to your organization.

Allocation Manager Dimensions

Due to platform limitation issues on DB2 UDB for OS / 390 and z/OS (the index size is limited to 255 characters) and Oracle (which requires 30 columns in an index) the predelivered index is on the first 10 dimensions. However, based on your requirements and the database platform, you can increase the maximum number of dimensions in the Allocation Manager to 28. The records that need to be modified for index changes include PF_AL_CALC_T, PF_AL_DIFF_T, PF_AL_DIV_T, PF_AL_SRC01_T, PF_AL_BAS02_T, and PF_AL_TOTAL_T.

To increase the number of dimensions:

1. Open each of the above records in the Application Designer.
2. Open the relevant subrecord.

   The delivered unique index is shown on the fields PF_AL_DIM1 to PF_AL_DIM10.
3. Depending on the number of dimensions you are adding, modify the index by adding the extra dimensions as keys.

   For example, if you want to use 15 dimensions then modify the index to include PF_AL_DIM11 through PF_AL_DIM15 and build the record. Alternatively, you can add a custom index on the table, using the Add Index feature.

Note. You can also apply these steps to Data Manager.

Mathematical Model for Data Manager and Allocation Manager Rules

The rules for Allocation Manager and Data Manager represent mathematical operations to be performed on the data you specify. Although Allocation Manger and Data Manager use different terminology, they perform similar calculations. For Allocation Manager, you must specify a source, basis, and target for the operation. For Data Manager you must specify two sources (source 1 and source 2) and the target.

The following table lists the Data Manager and Allocation Manager terms and how they relate.

Allocation Manager	*Data Manager*
Source	Source 1
Basis	Source 2

Allocation Manager	*Data Manager*
Target	Target

The *sources* of a rule use constraints to tell the Data Manager *what* is to be assigned and *what* basis to use. Use constraints to apply business rules to limit row selection. The *target* of a rule uses a datamap to tell the Data Manager *where* the rule should be located. The method that is applied to the rule determines the calculation that is performed.

Note. The explanation of the mathematical operations uses only Allocation Manger terms. Use the previous table to apply the Data Manager terms.

Understanding the Calculations Behind the Prorata and Spread Even Methods

The prorata and spread even methods perform the same calculation with the following exception: the prorata method uses measures from the basis table in the calculation, whereas the spread even method uses row counts from the basis to determine the ratio that is defined by basis measure ÷ basis total.

The prorata method performs the following calculation:

```
Source Measure * Basis Measure / Basis Total


Source Measure = Sum(Measures in source grouped by common dimensions and source⇒
 mapped dimensions)


Basis Measure = Sum(Measures in basis grouped by common dimensions and basis⇒
 mapped dimensions)


Basis Total = Sum(Measures in basis grouped by common dimensions only)
```

The spread even method performs the following calculation:

```
Source Measure * Basis Measure / Basis Total


Source Measure = Sum(Measures in source grouped by common dimensions and source⇒
 mapped dimensions)


Basis Measure = Sum(Count of basis rows grouped by common dimensions and basis⇒
 mapped dimensions)


Basis Total = Sum(Count of basis rows grouped by common dimensions only)
```

The following tables provide an example of the prorata method with one common dimension (unique dimension combinations).

Source:

Source Product	Source Amount
A	10
B	20

Basis:

Basis Product	Basis Channel	Basis Measure
A	X	10
A	Y	20
B	V	10
B	W	40

Target:

Target Product	Target Channel	Target Amount
A	X	3.33 (10 * 10/30)
A	Y	6.66 (10 * 20/30)
B	V	4 (10 * 10/50)
B	W	16 (20 * 40/50)

This table illustrates an example of the spread even method with one common dimension (unique dimension combinations).

Source

Source Product	Source Amount
A	10
B	20

Basis:

Basis Product	Basis Channel
A	X
A	Y
B	V

Basis Product	*Basis Channel*
B	W

Target:

Target Product	*Target Channel*	*Target Amount*
A	X	5 (10 * 1/2)
A	Y	5 (10* 1/2)
B	V	10 (20 * 1/2)
B	W	10 (20 * 1/2)

Understanding the Calculations Behind the Arithmetic Method

The arithmetic method performs the following calculation:

Source Measure [Operator] Basis Measure

Source Measure = Sum(Measures in source grouped by common dimensions and source⇒ mapped dimensions)

Basis Measure = Sum(Measures in basis grouped by common dimensions and basis⇒ mapped Dimensions)

Operator = [Addition (+), Subtraction (-), Multiplication (*), Division (/)]

Allocation Using Fixed Percentages

In Allocation Manager, you can divide a quantity by predetermined percentages and allocate those amounts by using the prorata method in combination with the fixed basis option.

In the calculation that is performed by the prorata method, the basis determines the ratio by which the source is divided. A ratio is another way to specify a percentage. By controlling the ratio, you can allocate the source amounts by specified percentages, or a fixed percentage.

```
Source Measure * (Basis Measure / Basis Total)
```

or

```
Source Measure * (%Percentage)
```

Warning! The prorata method always allocates 100 percent of the source. You must use the correct percentages when defining the fixed basis.

Setting Up Allocations

To set up allocations:

1. Complete your metadata setup, including datamap setups for the source, basis, and target definitions.

   This step is required.

   Note. PeopleSoft provides the *Datamap Wizard* to greatly simplify the creation of datamaps.

   The Datamap Wizard can be found using either of these navigation paths:

   EPM Foundation, Foundation Metadata, Metadata Wizards, Datamap Wizard

   EPM Foundation, Data Enrichment Tools, Allocation Manager, Datamap Wizard

   ---

   See Chapter 16, "Setting Up and Working with Metadata for the Operational Warehouse - Enriched," Setting Up Datamaps using the Datamap Wizard, page 367.

2. Define value objects.

   This step is optional.

3. Create indices on Allocation Manager temporary tables.

   This step is optional.

4. Define fixed dimensions.

   This step is optional if you do not use a fixed source or basis in your allocations.

5. Define allocation rules, including the method, source, basis, and target definitions.

   This step is required.

6. Create an Allocation Manager rule set that contains one or more allocation rules.

   This step is required.

7. Associate Allocation Manager rule set with a job.

   This step is required.

8. Process allocations using Run Allocations engine.

   This step is required.

Setting Up Value Objects

A value object provides descriptive information about fields and values. Value objects enhance the power of filters and can use constant strings to fill in target fields. Used in a filter, value objects play a role in the constraint's WHERE clause of a SQL command to enforce selection rules. There are two value object classes:

Meta Value A PeopleSoft variable that returns a field that is maintained by the system, such as the current system date, current system time, user ID, and so on, or a value that was entered as a parameter on the Data Manager run control prior to execution of Data Manager. Parameters include business unit, scenario ID, fiscal year, period, and so on.

Fixed Value A user-defined constant value. Examples of fixed values that are useful in the Data Manager include *0* (zero), and *N* (for yes/no fields).

Note. PeopleSoft delivers predefined value objects with PeopleSoft Enterprise Performance Management. However, you might need to define your own using the Value Object page.

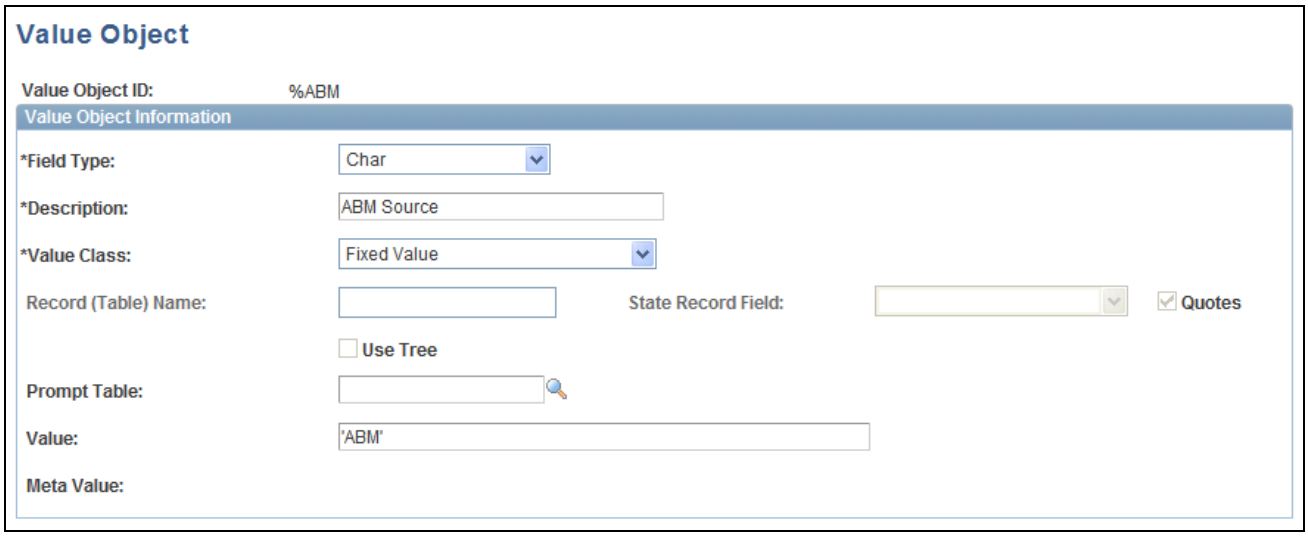
This section discusses how to define value objects.

Page Used to Set Up Value Objects

Page Name	*Definition Name*	*Navigation*	*Usage*
Value Object	MD_VALOBJ_TBL1	EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Value Object	Define or review value objects.

Defining Value Objects

Access the Value Object page (EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Value Object).



Value Object page

Value Object ID A unique identifier for the value object. All value objects are denoted in the system by a % at the beginning of the ID.

Field Type Select the field type. Choices are: *Amount, Basis Pts* (basis points), *Char* (character), *Date, Datetime,Number,* and *Rate.*

Value Class Select from the following:

Meta Value: Meta values are meta SQLs that retrieve system dates, user IDs, and engine run control parameters. If you select this option, you must specify the state record and state record field.

System Variables: System variables are values referencing system information (for example, the current date and time (%CurrentDateTime)).

Fixed Value: Fixed values are user-defined strings of text or numeric values that might be useful in specifying the definition of an assignment object or rule. These are usually constants.

If you select system variables or fixed values, the state record, state record field, quotes check box, and use tree check box are unavailable for entry. You must select the prompt table and enter a value.

State Record and **State Record Field** For *Meta Value,* enter the state record and state record field.

Quotes For *Meta Value,* select this check box for character strings. In most cases, you will not select this check box for numeric values.

Use Tree For *Meta Value,* select this check box if you want to use trees. The Prompt Table field changes to a Tree ID field and you can select your tree ID.

Prompt Table Use to select a prompt table as needed.

Value For *Meta Value,* the value field is automatically populated based on the state record and state record field selections. For example, if you select EPM_CORE_AET as your state record and BUSINESS_UNIT as your state record field, the value field is populated with : %BIND(EPM_CORE_AET.BUSINESS_UNIT).

For *Fixed Value ,* enter a character constant in quotes or a numeric constant without quotes.

For *System Variables,* enter the variable to use.

Meta Value Displays the meta value for any meta value objects.

Creating Indices on Allocation Manager Temporary Tables

Allocation Manager uses intermediate tables to temporarily store the results of an allocation calculation. When you perform allocations on large amounts of data, it is very important that indices are used correctly on all referenced temporary tables; this assures quality performance from the Allocation Manager. You should examine the indices on the temporary tables and ensure that they have been defined correctly, and are being used by the allocation SQL.

The intermediate tables used by Allocation Manager are:

- PF_AL_CALC_T
- PF_AL_DIFF_T
- PF_AL_DIV_T
- PF_AL_SRC01_T
- PF_AL_BAS02_T
- PF_AL_TOTAL_T

Due to platform limitation issues the prepackaged index is on the first 20 dimensions. Allocation Manager supports 30 dimensions. Depending on the number of dimensions you are using, modify the indices by adding the extra dimensions as keys, and rebuild the tables.

Defining Data Manager Rules

This section provides an overview of Data Manager rule methods, lists prerequisites, and discusses how to:

- Define a Data Manager rule.
- Define sources.
- Define the target.
- Review and define the source 1 and source 2 columns.
- Review the mapped common dimensions.
- Review the generated SQL.

Understanding Data Manager Rules and Methods

Data Manager rules use metadata to specify the source and the target tables for moving, aggregating, or creating multidimensional engine output.

Most rules have two sources: the measure to be assigned (for example Activity-Based Management indirect costs, revenue amounts, or inventory quantities) and the basis for the assignment (for example, per cent sales by region, product, or channel). Another datamap is selected as the target of the rule.

Data Manager rules define three things: source tables, targets, and the method that you want to apply to the data from the source tables as it moves to the target. These methods are:

- Copy method.
- Arithmetic method.
- GL Mapper method.
- Spread even method.
- Prorata method.
- Tree aggregation method.

Copy Method

The copy method moves data from the source to the target, but does nothing to the data. You would typically use multidimensional data as the source.

An example of this method would be the copy of REVENUE_F00, which is already attributed to the customer, product, and channel dimensions, to PF_JRNL_F00. In this move, no spreading of the revenue amount from the source to the target occurs.

In addition, this example shows that you can also use the copy method to generate aggregate amounts since more than one row of data might exist in source 1 for a customer/product/channel intersection. The copy method allows this type of aggregation by using the sum column on the Define Target page.

Source 1 datamap definition:

- DIMs: CUST_ID, PRODUCT_ID, CHANNEL_ID
- MSR: MONETARY_AMOUNT

Define Target page:

- Target fields of CUST_ID, PRODUCT_ID, CHANNEL_ID mapped to corresponding source 1 fields
- Sum check box selected for MONETARY_AMOUNT and mapped to source 1

Source 1 data:

Row	*CUST_ID*	*PRODUCT_ID*	*CHANNEL_ID*	*MONETARY_AMOUNT*
1	CU1111	PR111	CH11	1000.00
2	CU1111	PR111	CH22	2000.00
3	CU1111	PR222	CH11	1000.00
4	CU1111	PR222	CH11	2000.00

Row	CUST_ID	PRODUCT_ID	CHANNEL_ID	MONETARY_AMOUNT
5	CU1111	PR222	CH22	3000.00
6	CU2222	PR111	CH11	1000.00
7	CU2222	PR111	CH22	3000.00
8	CU2222	PR222	CH11	3000.00
9	CU2222	PR111	CH22	1000.00

Target results:

Row	CUST_ID	PRODUCT_ID	CHANNEL_ID	MONETARY_AMOUNT
1	CU1111	PR111	CH11	1000.00
2	CU1111	PR111	CH22	2000.00
3	CU1111	PR222	CH11	3000.00 (Aggregated)
4	CU1111	PR222	CH22	3000.00
5	CU2222	PR111	CH11	1000.00
6	CU2222	PR111	CH22	4000.00 (Aggregated)
7	CU2222	PR222	CH11	3000.00

Arithmetic Method

The arithmetic method applies an arithmetic operation (for example, add, subtract, multiply, or divide) to combine amounts in the sources that are grouped by common dimensions and source (or basis) mapped dimensions.

GL Mapper Method

The GL Mapper takes data from the general ledger tables and maps it to performance data. It populates the temporary table GLSTG. From GLSTG, you can use the copy method to move data directly to the performance journal or apply other methods to enrich your data.

For example, let's say that source 1 looks like this:

warehouse business unit	Account	Customer	Amount
CORP1	110	C1	1500 USD
CORP1	111	C1	1500 USD
CORP1	110	C2	1000 USD
CORP1	111	C2	1000 USD

If you map 100 percent of the amounts from source accounts 110 and 111 to performance account 100000, the target would look like this:

warehouse business unit	Account	Customer	Amount
CORP1	100000	C1	3000 USD
CORP1	100000	C2	2000 USD

Spread Even Method

The spread even method adds a dimension to your data, and spreads the fact amount evenly across the dimension. For example, you have revenue for three products as follows:

Source 1:

Product	Amount
P1	3000 USD
P2	2000 USD
P3	1000 USD

Source 2:

Customer
C1
C2

The target looks like this:

Product	*Customer*	*Amount*
P1	C1	1500 USD
P1	C2	1500 USD
P2	C1	1000 USD
P2	C2	1000 USD
P3	C1	500 USD
P3	C2	500 USD

The amounts for product are spread evenly across the customers.

Prorata

The prorata method adds an additional dimension to your data and divides the amount across that dimension based on the percentage of the total amount. For example, you have revenue for three products as follows:

Source 1:

Product	*Amount*
P1	3000 USD
P2	2000 USD
P3	1000 USD

You also want to attribute that revenue to customers. The revenue by customers is:

Source 2:

Customer	*Amount*
C1	4000 USD
C2	2000 USD

The target looks like this:

Product	*Customer*	*Amount*
P1	C1	2000 USD
P1	C2	1000 USD

Product	*Customer*	*Amount*
P2	C1	1333.33 USD
P2	C2	666.67 USD
P3	C1	666.67 USD
P3	C2	333.33 USD

The amounts for the products are prorated based on the percentage of total for each customer (C1 has two-thirds, or 66.67 percent, and C2 has one-third or 33.33 percent).

Tree Aggregation

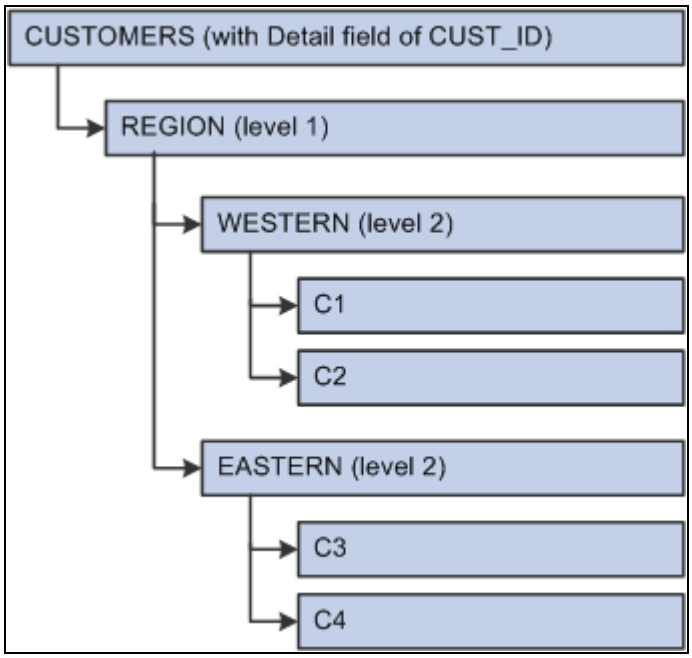
In the tree aggregation method, measures are aggregated based on a tree hierarchy. The node names act as the dimension. For example, assume you have the following data for individual customer IDs.

Source 1:

Customer	*Product*	*Channel*	*Amount*
C1	P1	CH1	1000
C1	P1	CH1	1000
C1	P2	CH2	2000
C2	P1	CH1	1000
C2	P2	CH2	2000
C3	P2	CH2	2000
C4	P1	CH1	1000
C4	P2	CH2	2000
C4	P2	CH2	2000

Assume you have the following customer location tree structure.

Source 2:



Sample customer tree

The system rolls up the source data (customer IDs) to the next level up (regions) summarizes it, and reduces the number of rows. In the target table, the customer ID is now the tree node at the desired roll-up level.

This is how the target looks:

Customer	***Product***	***Channel***	***Amount***
WESTERN	P1	CH1	3000
WESTERN	P2	CH2	4000
EASTERN	P1	CH1	1000
EASTERN	P2	CH2	6000

Prerequisites

Before setting up your Data Manager rules, you must:

- Complete set up of your metadata (tablemaps, datamaps, and constraints).
- Define any value objects on the Value Object page.

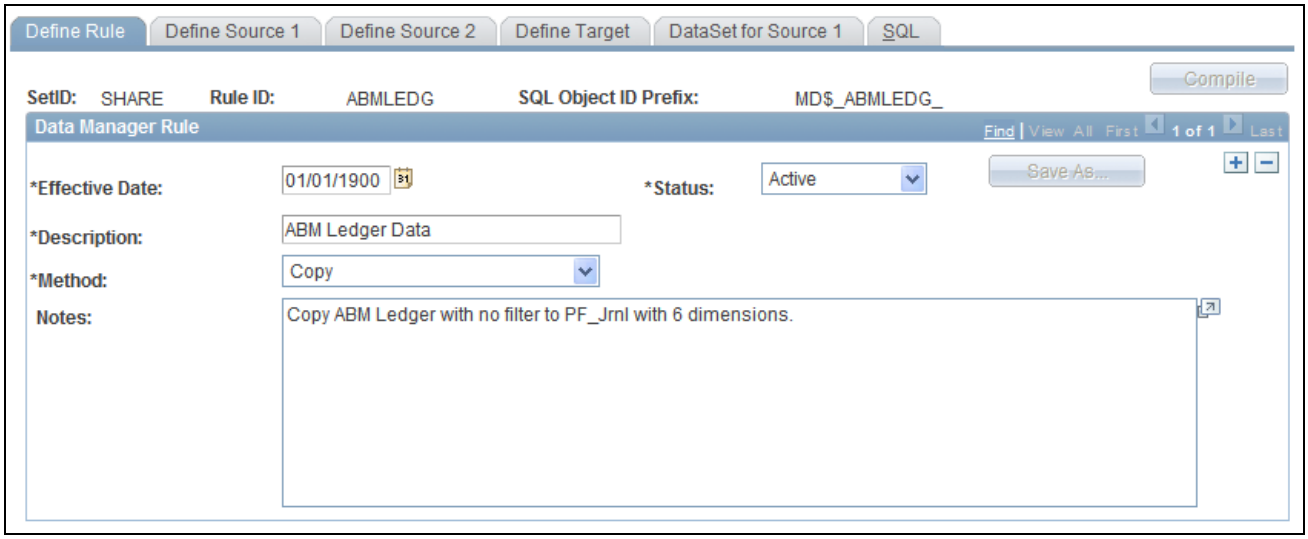
See Chapter 16, "Setting Up and Working with Metadata for the Operational Warehouse - Enriched," page 339 and Chapter 22, "Using Data Enrichment Tools," Setting Up Value Objects, page 548.

Pages Used to Set Up Data Manager Rules

Page Name	*Definition Name*	*Navigation*	*Usage*
Data Manager Rules - Define Rule	MD_RULE_TBL1	EPM Foundation, Data Enrichment Tools, Data Manager, Rules, Define Rule	Define a data manager rule, including the method you intend to use.
Data Manager Rules - Copy Rule	MD_RULE_COPY	Click on the Save As button on the Define Rule page.	Copy a Data Manager rule and save it under a new rule name.
Data Manager Rules - Define Source 1	MD_RULE_SCR1	EPM Foundation, Data Enrichment Tools, Data Manager, Rules, Define Source 1	Define source 1 information.
Data Manager Rules - Define Source 2	MD_RULE_SCR2	EPM Foundation, Data Enrichment Tools, Data Manager, Rules, Define Source 2	Define source 2 information.
Data Manager Rules - Define Target	MD_RULE_TBL3	EPM Foundation, Data Enrichment Tools, Data Manager, Rules, Define Target	Define the target for the rule and the target mapping.
Data Manager Rules - DataSet for Source 1	MD_DATASET1_DEFN	EPM Foundation, Data Enrichment Tools, Data Manager, Rules, DataSet for Source 1	Review and define the source 1 fields and measures that are mapped to target fields.
Data Manager Rules - DataSet for Source 2	MD_DATASET2_DEFN	EPM Foundation, Data Enrichment Tools, Data Manager, Rules, DataSet for Source 2	Review and define the source 2 fields and measures that are mapped to target fields. This page displays the prorata and spread even methods only.
Data Manager Rules - DataSet for Common Dimensions	MD_DATASET3_DEFN	EPM Foundation, Data Enrichment Tools, Data Manager, Rules, DataSet for Common Dimensions	Review the mapped common dimensions and prorata measure.
Data Manager Rules - SQL	MD_RULE_TBL4	EPM Foundation, Data Enrichment Tools, Data Manager, Rules, SQL	Review the generated SQL for the Data Manager rule.

Defining a Data Manager Rule

Access the Data Manager Rules - Define Rule page (EPM Foundation, Data Enrichment Tools, Data Manager, Rules, Define Rule).



Define Rule page

Method

Select the method for the rule you are defining. Options are: *Arithmetic*, *Copy*, *GL Mapper*, *Prorata*, *Spread Even*, and *Tree Aggregation.*

See Chapter 22, "Using Data Enrichment Tools," Understanding Data Manager Rules and Methods, page 551.

Note. If the method is changed after the target fields have been populated, some fields might be reset to blank. This occurs if the current method has a source 2 and the new method does not. All source 2 fields that were mapped in the Data Manager Rules component are reset. The system displays a warning message.

For a method of *Arithmetic*, select the operation to use: *Addition (+)*, *Division (/)*, *Multiplication (*)*, and *Subtraction (–).*

For a method of *Prorata*, the system prorates measures in source 1 across all common dimensions based on source 2 numbers.

For a method of *Spread Even*, the system spreads source 1 measures evenly across all common dimensions based on source 2 numbers.

For a method of *Tree Aggregation*, the system copies data elements from the source to the target based on the source 2 tree structures.

Save As

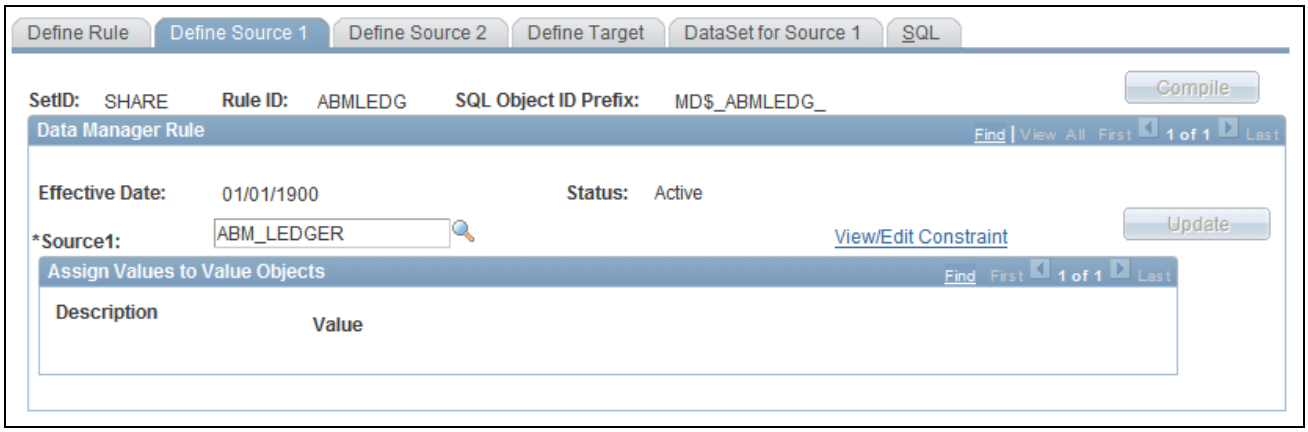
In correction mode, click the Save As button to access the Copy Rule page on which you can enter a new rule ID to which you want to copy the rule information. Enter an effective date as today's date. Click OK to save the changes and return to the Define Rule page.

Compile

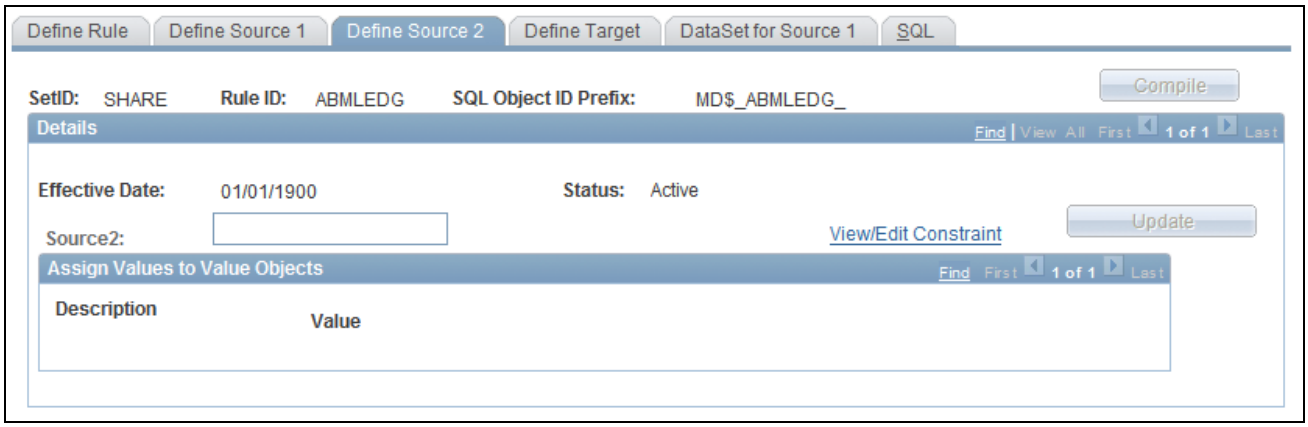
Click the Compilebutton on this page or other pages in the component to build the Data Manager rule. Review the generated SQL on the Data Manager Rules - SQL page.

Defining Sources

Access the Data Manager Rules - Define Source 1 page (EPM Foundation, Data Enrichment Tools, Data Manager, Rules, Define Source 1) and the Data Manager Rules - Define Source 2 page (EPM Foundation, Data Enrichment Tools, Data Manager, Rules, Define Source 2).



Define Source 1 page



Define Source 2 page

Source 1 Select the constraint to tell the Data Manager what to assign.

The constraints you define can limit the row selection. You can also define constraints to have no criteria, in which case they pull all the data. Most rules have two sources, with the exception of the copy and GL Mapper methods, which only require one source. Source 1 contains the measure to be assigned (for example, the Activity-Based Management costs), while source 2 contains the basis for the assignment (for example, spreading those costs either evenly or prorating them over customers).

For the GL Mapper method, the source constraint must be *GLM_LEDGER_ALL*. This constraint must be built on the primary table of the LEDGER_VW tablemap.

Note. To review the primary table of the LEDGER_VW tablemap, click the View/Edit Constraint link to navigate to the Constraint page. On the Constraint page, click the View/Edit Datamap link.

For the tree aggregation method, a source 1 mapping field displays next to the Source 1 field. The Data Manager uses this field in conjunction with the tree node field to determine the join criteria in the SQL.

Source 2 Select the constraint to tell the Data Manager what basis to use.

For the tree aggregation method, select the tree you want to use for aggregation. When you select the tree, the Tree Viewer opens and enables you to drill down to the tree level you require. You can also review the tree node information. Click the Select button on the Tree Viewer page to select the tree level.

Note. You do not need to specify a source 2 for the copy or GL Mapper methods.

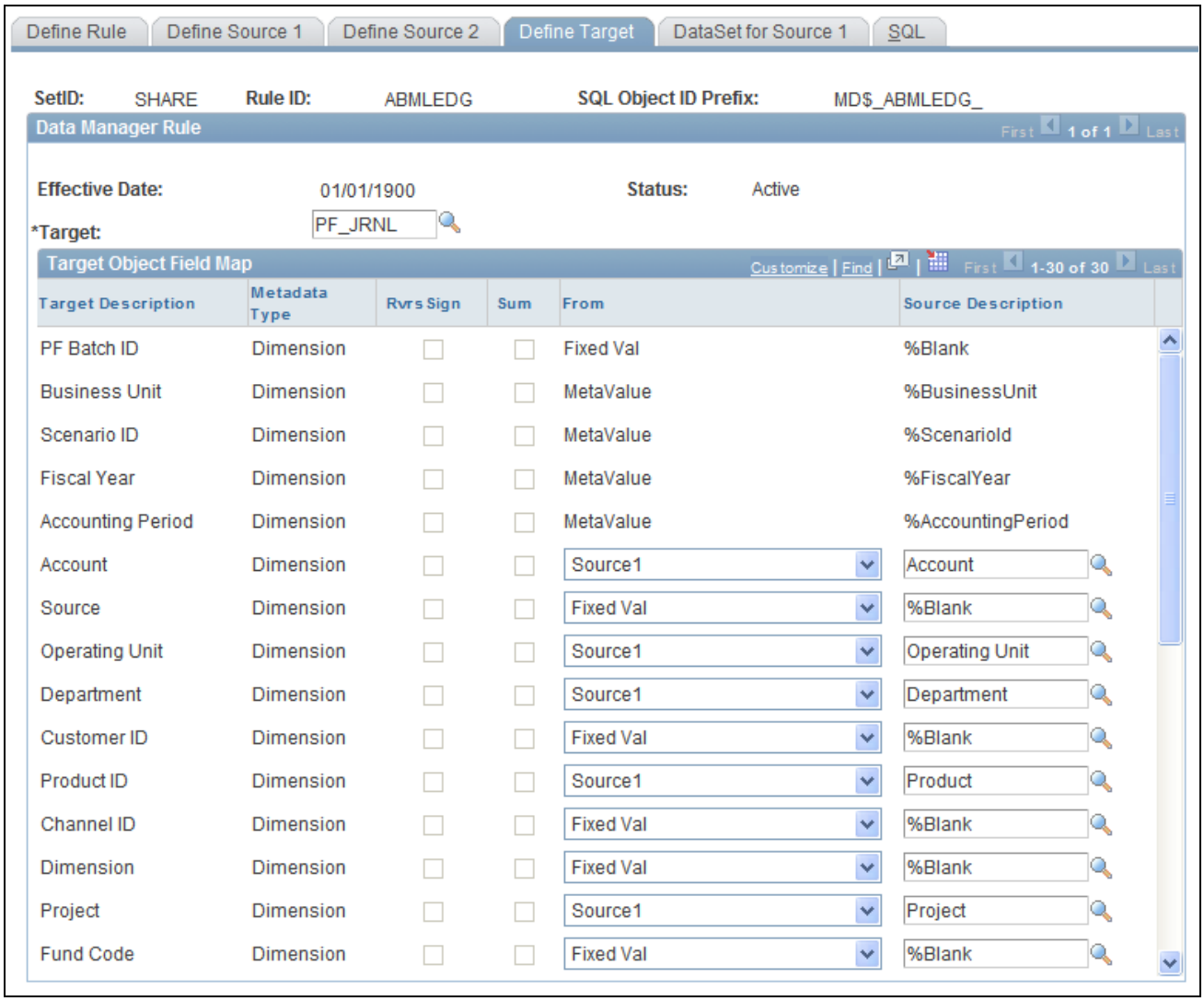
View/Edit Constraint Click the link to view or edit the selected constraint. This option transfers you to the Constraint page for that constraint.

Update Click the Update button to view an updated list of the source 1 or source 2 assignment values. You must be in correction mode to click this button.

Note. If you change a source constraint after the populating the target fields, the target fields that were mapped to source fields will not be lost so long as the datamap is the same.

Defining the Target

Access the Data Manager Rules - Define Target page (EPM Foundation, Data Enrichment Tools, Data Manager, Rules, Define Target).



Define Target page

Target

Select the datamap to use as the target for the processed data.

The system populates the Target Object Field Map grid with target fields based on the datamap you select. The system handles the mapping of all the fields except for the department and product fields, which you can only map to the source 1 department or product, a fixed value, or a meta value.

For the GL Mapper method, the target is *GLSTG*.

Warning! For the prorata and spread even methods, each target field must map to a source 1 field, source 2 field, or value object. The system issues a warning message if any dimensions exist on the datamap that are not mapped to a target field.

Prorata Measure

For a method of *Prorata*, select the measure to use for the prorata calculation. Only source 2 fields that are defined as measures on the datamap display in the prompt list.

Measure For a method of *Arithmetic,* select the measure to use. The values in the list box are based on the selected source 2 constraint.

Target Description and **Metadata Type** The system populates these fields based on the datamap you select as the target. The target description comes from the datamap definition.

RvrsSign (reverse sign) Select to reverse the sign of the amount when the data is copied to the target table. Check boxes are available in this column only for target fields that are defined as a measure on the target datamap.

Sum For a method of *Copy,* select this check box to total the amounts when the source 1 data is copied to the target table. Check boxes are available in this column only for target fields that are defined as a measure on the target datamap.

From This field can be:

Source 1: Data is copied from the source 1 table to the target field.

Fixed Value: A constant value is inserted into the target field.

Meta Value: A meta value is inserted into the target field.

Sys Var: (system variable) A system variable is inserted into the target field.

Note. For the GL Mapper method, you can only specify the From field for the department and product ID.

Source Description If the value in the From field is *Source 1* or if the value is mapped from source 1, a list box of the fields that were defined in source 1 is available.

If the value in the From field is *Source 2* or if the value is mapped from source 2, a list box of the fields that were defined in source 2 is available.

If one of the value object types is specified for the value in the From field, the drop-down list box contains the objects that correspond to the specified value object class.

Prorata For a method of *Prorata,* select this check box for any measures you want to use for the proration. You must select at least one measure field for prorata.

Arithmetic For a method of *Arithmetic,* select this check box for a field in which you want to use in the arithmetic operation that is specified on the Data Manager Rules - Define Rule page.

Target Definition for the Copy Method

When you are setting up the target field map for the copy method:

- If you select a target measure field that is mapped to a source 1 field for a sum operation, all other measures that are mapped to source 1 fields must also be selected for a sum operation.
- If you select the sum check box for any of the fields, you can map only key source 1 fields.

- If you do not select the sum check box for any of the fields, you can map any source 1 fields to any target fields.

  This might result in duplicate data when you run the Data Manager engine. This is because you are not grouping the data by the key values.

- You can use expressions for your mappings.

Target Definition for the Prorata Method

When you are setting up the target field map for the prorata method:

- Only three mapped measures are allowed.
- The source 1 and source 2 datamaps must both have at least one key field that is marked as a dimension.
- All non-measure target fields can be mapped only to source 1 or source 2 fields that are marked as dimensions (keys) on the datamap.
- All measure target fields can only be mapped to source 1, fixed value, or meta value fields.
- At least one measure target field must be marked for prorata.
- If a target field is mapped to a source 1 field, it must be marked for prorata.

Target Definition for the Spread Even Method

When you are setting up the target field map for the spread even method:

- Only three mapped measures are allowed.
- The source 1 and source 2 datamaps must both have at least one key field that is marked as a dimension.
- All non-measure target fields can only be mapped to source 1 or source 2 fields that are marked as dimensions (keys) on the datamap.
- All measure target fields can only be mapped to source 1, fixed value, or meta value fields.

  Data Manager uses all the source 1 mappings for the spread even calculation.

- At least one measure target field must be mapped to a source 1 field.

Target Definition for Tree Aggregation Method

When you are setting up the target field map for the tree aggregation method:

- Each target field must have a source 1 field, tree node, or value object to which it is mapped.
- Only one target field can be mapped to the source 2 tree node.

  This tree node field is used in conjunction with the source 1 mapping field for join criteria in the SQL.

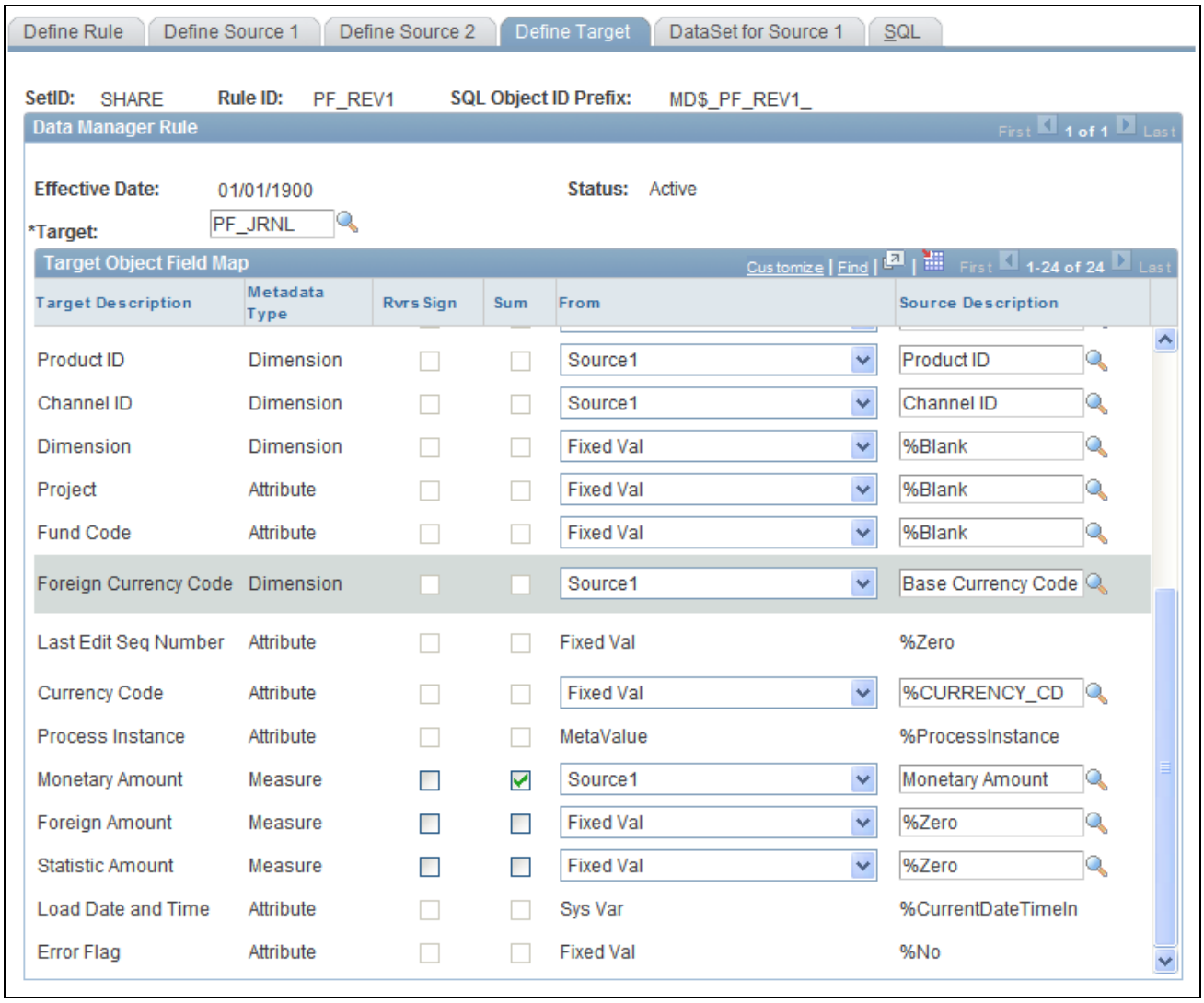
- You must have at least one non-measure target field mapped to source 1.

- If the target field is mapped to source 2, you can select from all the fields for that source.

  If a target field is mapped to source 2, the only selection available is *Tree Node.*

Defining Foreign Currency

You must always provide values for the Foreign Currency Code in the target definition. Even if you are using United States dollars as your base currency, you must populate the Foreign Currency Code in the target definition. For example, if you use United States dollars as your base currency, you must map *Source1* to *Base Currency Code* (as the following graphic demonstrates).



Define Target page - Foreign Currency Code field

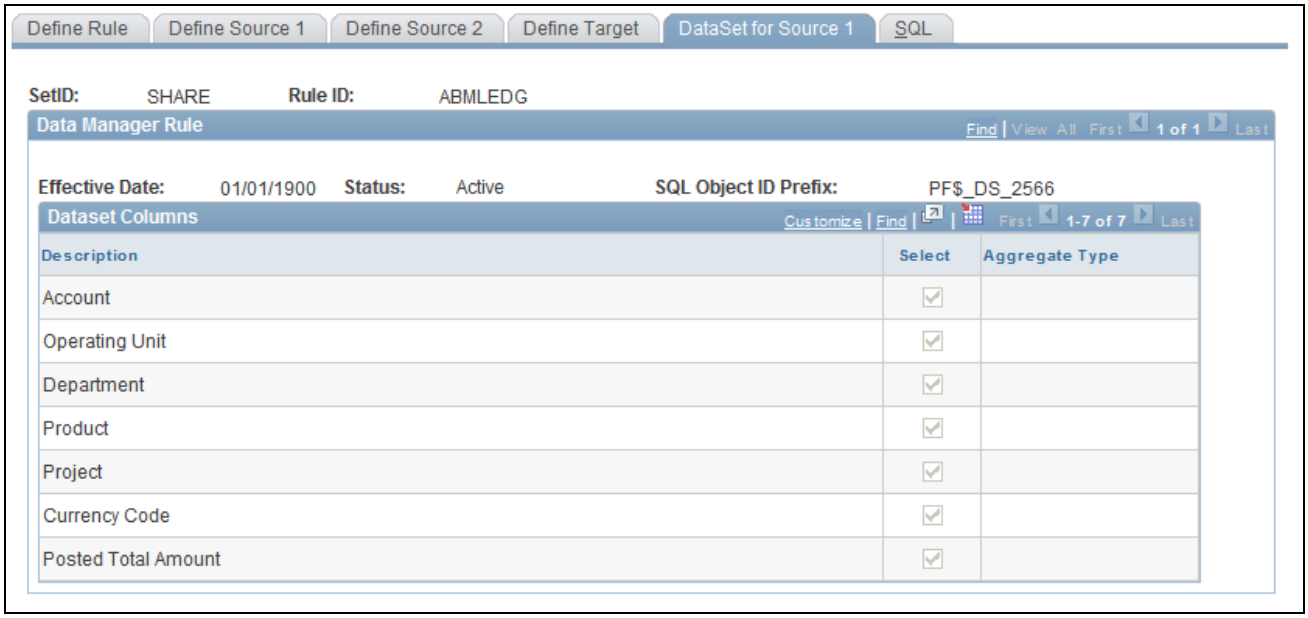
If you do not populate the Foreign Currency Code in the target definition, unexpected results may occur when the PF_POST application engine runs.

See Also

Chapter 22, "Using Data Enrichment Tools," Setting Up Value Objects, page 548

Reviewing and Defining the Source 1 and Source 2 Columns

Access the Data Manager Rules - DataSet for Source 1 page (EPM Foundation, Data Enrichment Tools, Data Manager, Rules, DataSet for Source 1) and Data Manager Rules - DataSet for Source 2 page (EPM Foundation, Data Enrichment Tools, Data Manager, Rules, DataSet for Source 2).



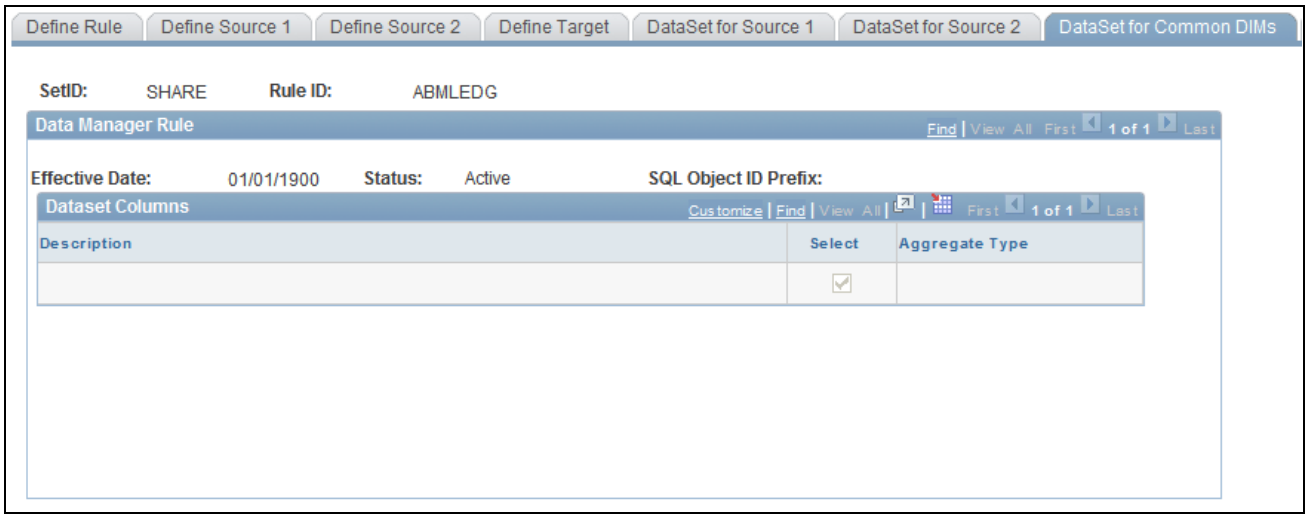
DataSet for Source 1 page

Select Select this check box to include the dimension or measure.

Note. The DataSet for Source 2 page does not display for the copy or GL Mapper method.

Reviewing the Mapped Common Dimensions

Access the Data Manager Rules - DataSet for Common DIMs page (EPM Foundation, Data Enrichment Tools, Data Manager, Rules, DataSet for Common Dimensions).



DataSet for Common DIMs page

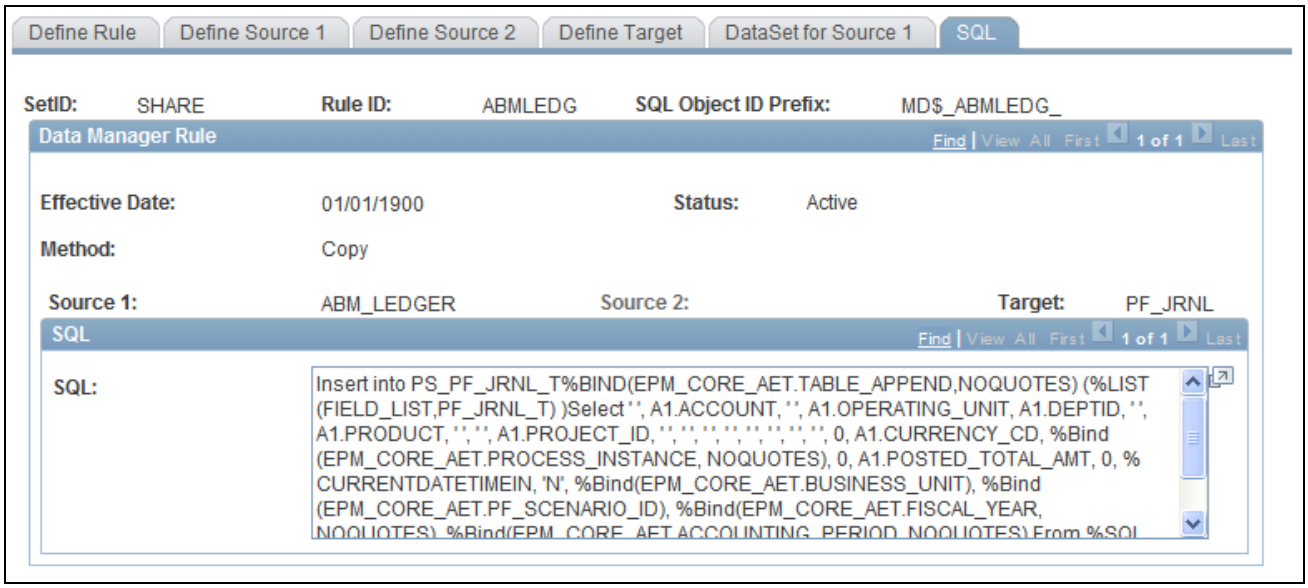
This page shows the common dimensions that exist between source 1 and source 2. A dimension is common if it has a matching description in the source 1 and source 2 datamap fields.

For the prorata method, the page also shows the prorata measure marked as *Sum*.

Note. Common dimensions are very important. They determine the join criteria for the method. The criteria are based on the datamap, so always review this page to ensure the results are calculating as you expect.

Reviewing the Generated SQL

Access the Data Manager Rules - SQL page (EPM Foundation, Data Enrichment Tools, Data Manager, Rules, SQL).



SQL page

Once you have compiled a rule using the Compile button, review the generated SQL statement.

All %Bind and %SQL objects are resolved at run-time. There is only one SQL statement that is generated for each method.

A copy of this SQL is stored in the SQL repository with the SQL object ID prefix that is shown in this component. A sequential number for each SQL statement completes the SQL object name for the repository.

Defining Data Manager Rule Sets

You must define a Data Manager rule set for any Data Manager rules that you want to process. You can also use rule sets to combine multiple rules in one run of the Data Manager engine. Rules in a rule set are run in the order shown on the Data Manager Rule Sets page. The first rule must complete successfully before the second rule starts. The system places the data from the first rule in a temporary table (assigned by the engine) that will be the source of the second rule, and so on.

Note. A rule must be in a rule set, even if only one rule is processed.

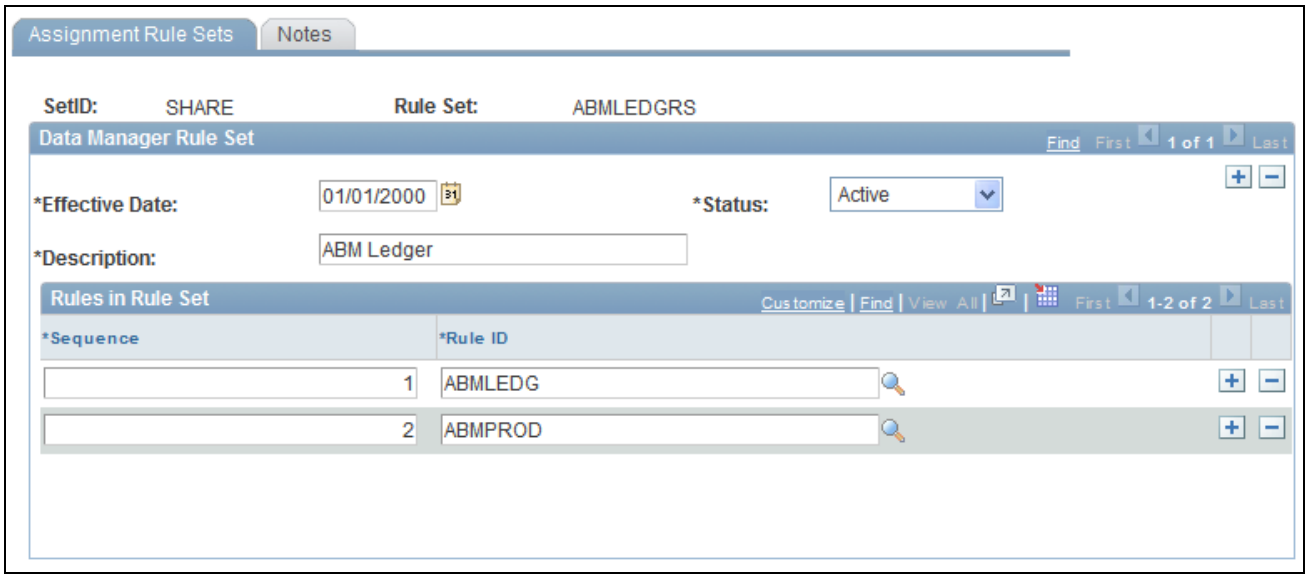
This section discusses how to set up Data Manager rule sets.

Pages Used to Define Data Manager Rule Sets

Page Name	*Definition Name*	*Navigation*	*Usage*
Assignment Rule Sets	MD_RULESET_TBL1	EPM Foundation, Data Enrichment Tools, Data Manager, Rule Set, Assignment Rule Sets	Set up Data Manager rule sets by defining the rules that are to be included in the rule set.
Assignment Rule Sets - Notes	MD_RULESET_TBL2	EPM Foundation, Data Enrichment Tools, Data Manager, Rule Set, Assignment Rule Sets	Describe the purpose of this rule set.

Setting Up Data Manager Rule Sets

Access the Assignment Rule Sets page (EPM Foundation, Data Enrichment Tools, Data Manager, Rule Set, Assignment Rule Sets).



Assignment Rule Sets page

RuleSet A unique identifier for this rule set definition.

Sequence Enter a number, such as 100, for the first rule ID in the rule set. The next rule ID to run would have a sequence of 200. The actual sequence number is not important; it represents the order in which you want to process rules. The sequence must be unique.

Rule ID Select the Data Manager rule to include in the rule set. You create the rules using the Data Manager Rules component.

Click the Notes tab to enter a more detailed description of this rule set.

Renaming Data Manager Rules

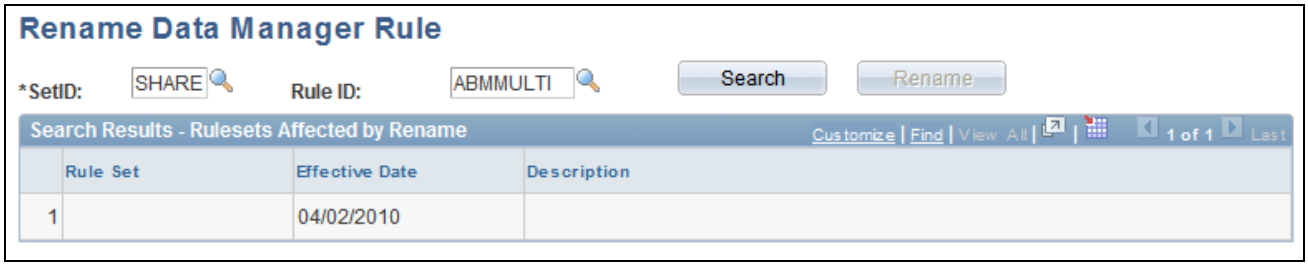
This section discusses how to rename a Data Manager rule.

Page Used to Rename Data Manager Rules

Page Name	*Definition Name*	*Navigation*	*Usage*
Rename Data Manager Rule	PF_MD_RULE_RENAME	EPM Foundation, Data Enrichment Tools, Data Manager, Rename Rules, Rename Data Manager Rule	Rename existing rules for modification.

Renaming a Data Manager Rule

Access the Rename Data Manager Rule page (EPM Foundation, Data Enrichment Tools, Data Manager, Rename Rules, Rename Data Manager Rule).



Rename Data Manager Rule page

SetID and **Rule ID** Select the setID and rule ID for the rule you want to rename.

Search Click the Search button. The search populates the grid with the names of the Data Manager rule sets that is affected by this rename.

Rename To proceed with the renaming process, click the Rename button and enter a new name for the rule. Every instance in which the original rule name is used is changed to the new name.

Associating Data Manager Rule Sets With a Job

The next step in the Data Manager process is associating the rule set with a Data Manager job. Each unique run of the Data Manager engine is given a job ID. You must establish an association to the rules and the actual job that will run them on the Job Association page. When the jobstream that contains the defined Data Manager job ID runs, it uses the setup on the Job Association page to find the Data Manager rule set to execute.

Note. Each time that you run an EPM engine, it must have a unique job ID. These jobs must then be run in a jobstream.

This section discusses how to define rule sets to be run for a given job.

Prerequisites

Before you can associate rule sets with job, you must:

- Set up the Data Manager job ID.
- Set up a jobstream for the job ID.

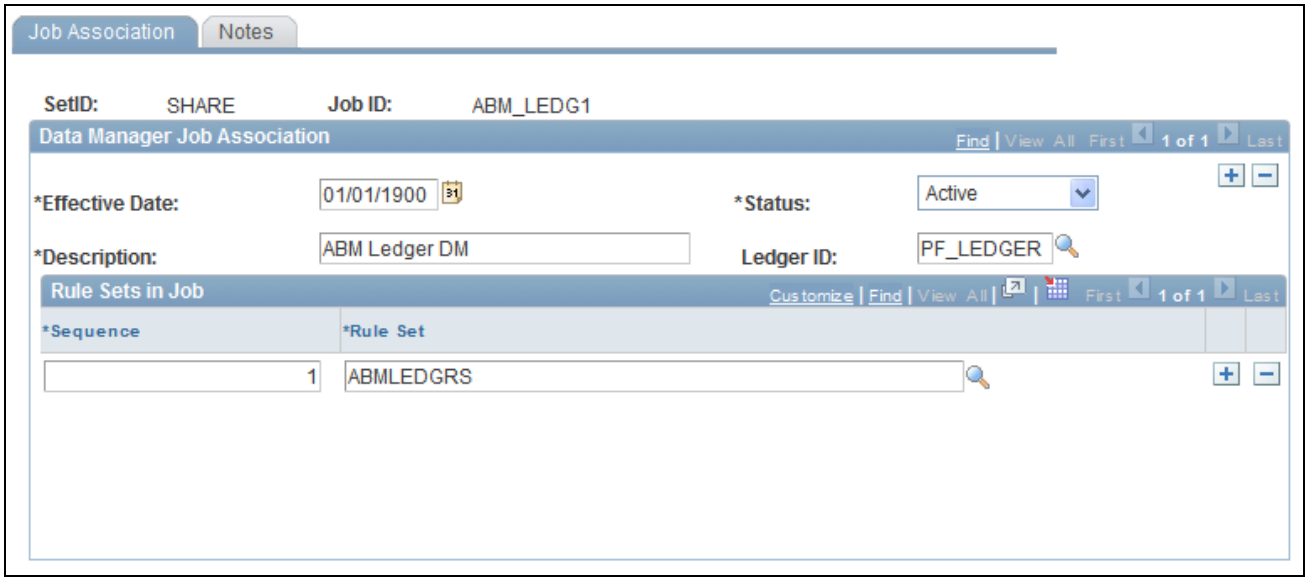
See Chapter 20, "Streamlining Processing with Jobstreams," page 461.

Pages Used to Associate Data Manager Rule Sets With a Job

Page Name	Definition Name	Navigation	Usage
Job Association	MD_JOB_RULESET_TBL	EPM Foundation, Data Enrichment Tools, Data Manager, Job Association, Job Association	Define rule sets to be run for a given job ID. The order in which these rule sets are run is also defined.
Job Association - Notes	MD_JOB_RULESET_TB2	EPM Foundation, Data Enrichment Tools, Data Manager, Job Association, Notes	Describe the job association.

Define Rules Sets to be Run for a Given Job

Access the Job Association page (EPM Foundation, Data Enrichment Tools, Data Manager, Job Association, Job Association).



Job Association page

Job ID Select the job ID. The job ID was created in the job metadata and represents a unique instance of a Data Manager engine in a jobstream.

Ledger ID Select a ledger ID. This is required for the Profit Manager.

See Chapter 21, "Setting Up and Using Profit Manager," Processing and Posting Journals, page 516.

Sequence	Enter a number, such as 100, for the first rule set in the job association. The next rule set to run would have a sequence of 200. The actual sequence number is not important; it represents the order in which you want to process rule sets. The sequence must be unique.
Rule Set	Select the rule set or rule sets to include in the job. You create rule sets on the Data Manger Rule Set - Assignment Rule Sets page.

Click the Notes tab to enter a more detailed description of this job association.

Running the Data Manager Engine

As a last step, you run the Data Manager engine in a jobstream. As you run the jobstream:

- Use the Process Monitor to verify the status of your job.
- Review the results by querying the target table for the Data Manager rules that you have run.
- Review any error messages using the Error Messages component.

See Also

Chapter 20, "Streamlining Processing with Jobstreams," Running Jobstreams, page 485

Chapter 20, "Streamlining Processing with Jobstreams," Viewing Engine Messages, page 494

Defining Allocation Manager Rules

This section provides an overview of allocation rules setup, fixed source and fixed basis options in allocations, lists prerequisites, and discusses how to:

- Determine the list of dimensions to use.
- Create a list of sources.
- Define an Allocation Manager rule.
- Define the allocation rule source.
- Define the allocation rule basis.
- Review common dimensions.
- Define the target.
- Assign residual amounts.
- Specify offset target information.

- Define allocation template.

Understanding Allocation Rule Setup

The following list represents required and optional tasks necessary to set up allocation rules.

1. Define datamaps for your source, basis, and target.

   This step is a required prerequisite.

   Note. PeopleSoft provides the *Datamap Wizard* to greatly simplify the creation of datamaps.

   The Datamap Wizard can be found using either of these navigation paths:

   EPM Foundation, Foundation Metadata, Metadata Wizards, Datamap Wizard

   EPM Foundation, Data Enrichment Tools, Allocation Manager, Datamap Wizard

   ---

   See Chapter 16, "Setting Up and Working with Metadata for the Operational Warehouse - Enriched," Setting Up Datamaps using the Datamap Wizard, page 367.

2. Select fixed dimensions to use with a fixed source or basis.

   This step is optional if you are not using a fixed source or basis.

3. Select DataMaps for the allocation source, basis, and target.

   This step is required.

4. Define an allocation rule and select the calculation method.

   This step is required.

5. Define the allocation rule source.

   This step is required.

6. Define the allocation rule basis.

   This step is required.

7. Review all common dimensions between the source and the basis.

   This step is required.

8. Define the allocation rule target, including the allocation mappings.

   This step is required.

9. Assign post-allocation residual amounts.

   This step is optional.

10. Specify offset target information.

    This step is optional.

With the exception of the first list item above, you use the Allocation Manager Rules component to set up all remaining tasks.

Note. You must set up the DataMaps and constraints you want to use in the allocation prior to creating an allocation rule.

See Chapter 22, "Using Data Enrichment Tools," Defining Allocation Manager Rules, page 572 and Chapter 16, "Setting Up and Working with Metadata for the Operational Warehouse - Enriched," Setting Up Datamaps, page 363.

Streamlining Allocation Rule Setup with the Allocation Template

PeopleSoft EPM provides an *allocation template* that enables you to predefine a set of source, basis, and target DataMaps that you can later associate with any allocation rule, thereby saving time and ensuring DataMap consistency among various users setting up allocation rules.

When you define an allocation template you can associate it with an allocation rule using the *Template* field on the Method page, which is a part of the Allocation Manager Rules component. Once you select a template on the Method page, the appropriate source, basis, and target DataMaps automatically populate the Source, Basis, and Target pages.

See Chapter 22, "Using Data Enrichment Tools," Defining an Allocation Template (Optional), page 592.

Understanding Allocation Manager Fixed Source and Fixed Basis Options

Allocation Manager gives you the flexibility of adding fixed source and basis values in an allocation rule without referencing a database table. For example, you may only have a few dimension values that you want to use in a rule, like a source amount of $100,000 that you want to allocate to the accounts in the Basis table. Or, you may want to perform an arithmetic allocation and multiply all values by a fixed rate of 0.25. In these cases, it is much simpler to input the value in the rule, without having to create a database table.

You use datamap metadata to define the tables that contain the source, basis, and target for your allocation in the Allocation Manager rule. However, in some cases you might not have source or basis data stored in tables in your database. To accommodate this situation, we deliver a set of metadata (the FIX_BASIS datamap, filters and constraints PF_AL_FIX_SOURCE , PF_AL_FIX_BASIS) that point to the same fixed source and basis table (PS_FP_AL_FIX_BASIS). You can tailor this table by adding your organization's dimensions to it on the Fixed Dimensions page. Use the Fixed Dimension page to list the dimensions and assign prompt tables and trees and select values for each dimension.

Note. Fixed source and basis is used with the Allocation Manager only. It is available with all allocation methods except period-based allocation.

Note. Before you create an Allocation Manager rule to associate with a setID, create a PF_AL_FIX_BASIS and PF_AL_FIX_SOURCE filter and constraint for that setID. You can model the setup for the PF_AL_FIX_BASIS and PF_AL_FIX_SOURCE filter and constraint using the sample data under the SHARE setID.

Calculating Offset Target Information (Optional)

Allocation Manager enables you to calculate offsets for the target balance. Offset accounts aid in balancing source transaction and budget entries. You can use offsets to balance ledgers. You can create either single or multiple offsets. A single offset allows a single target entry for the entire offset and multiple offsets create identical set of target entries with the target measure values negated. Also, a separate offset entry can be created for each of the residual amounts. You must ensure at least one key value is different from the residual entry in order to avoid collisions. The same is true if a single offset entry is selected.

The following is an example of an arithmetic method offset allocation account:

Source		***Basis***		***Target***	
Department	Amount	Department	Unit	Department	Amount
CORP	1000	DIV1	0.1	DIV1	100
		DIV2	0.1	DIV2	100
		DIV3	0.2	DIV3	200
				OFST (Offset Account)	=400

In this example, the offset account is equal to 400, which is the same total for Divisions (DIV) 1, 2, and 3.

Prerequisites

Before setting up your Allocation Manager rules, you must:

- Complete the setup of your metadata (tablemaps, datamaps, and constraints).
- Define any value objects on the Value Object page.

See Chapter 16, "Setting Up and Working with Metadata for the Operational Warehouse - Enriched," page 339 and Chapter 22, "Using Data Enrichment Tools," Setting Up Value Objects, page 548.

Pages Used to Define Allocation Manager Rules

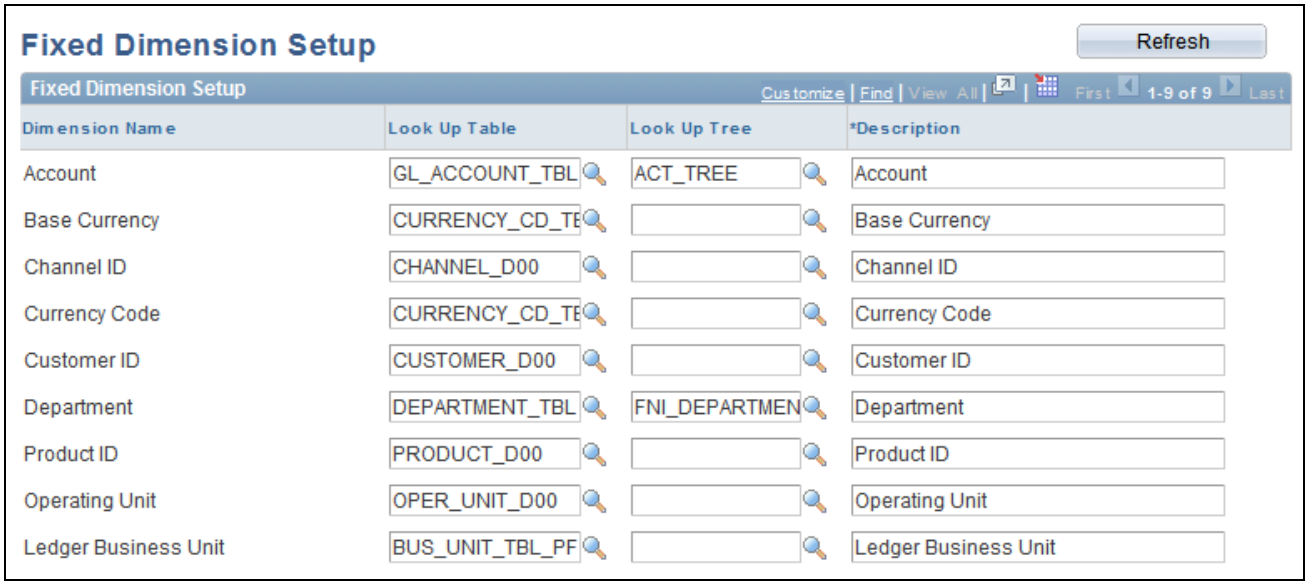
Page Name	***Definition Name***	***Navigation***	***Usage***
Fixed Dimension Setup	PF_AL_DIM_TBL	EPM Foundation, Data Enrichment Tools, Allocation Manager, Fixed Basis Dimensions, Fixed Dimension Setup	Determine the list of dimensions that you can use with fixed source and basis.

Page Name	Definition Name	Navigation	Usage
Allocation DataMap Setup	PF_AL_DM_TBL	EPM Foundation, Data Enrichment Tools, Allocation Manager, DataMap Setup, Allocation DataMap Setup	Create a list of sources for the source, basis, or target and specify dimensions for your datamap. The purpose of this page is to reduce the of number datamaps from which to choose. In addition, you can give the datamaps more intuitive names. Add as many datamaps as necessary.
Select Dimensions	PF_AL_SELECT_DIM	Click the  Select Dimensions button on the Allocation DataMap Setup page	Select the dimension metadata type.
Rules - Method	PF_AL_METHOD_DEFN	EPM Foundation, Data Enrichment Tools, Allocation Manager, Rules, Method	Define an Allocation Manager rule and select the method.
Rules - Copy Rule	PF_AL_RULE_COPY	Click on the Save As button on the Allocation Manager Rules - Method page.	Copy Allocation Manager rules.
Rules - Source	PF_AL_SOURCE_DEFN	EPM Foundation, Data Enrichment Tools, Allocation Manager, Rules, Source	Define the allocation rule sources.
Fixed Source Dimension Setup	PF_AL_USE_FSRC_DIM	Click on the Choose Fixed Source Dimension link on the Allocation Manager Rules - Source page.	Select which dimensions for a fixed source.
Rules - Basis	PF_AL_BASIS_DEFN	EPM Foundation, Data Enrichment Tools, Allocation Manager, Rules, Basis	Define the allocation rule basis.
Fixed Basis Dimension Setup	PF_AL_USE_FIX_DIM	Click on the Choose Fixed Basis Dimension link on the Allocation Manager Rules - Basis page.	Select which dimensions for a fixed basis.
Rules - Source to Basis Relationship	PF_AL_COM_DIM_DEFN	EPM Foundation, Data Enrichment Tools, Allocation Manager, Rules, Source to Basis Relationship	Review all common dimensions between the source and the basis.

Page Name	Definition Name	Navigation	Usage
Rules - Target	PF_AL_TGT_DEFN	EPM Foundation, Data Enrichment Tools, Allocation Manager, Rules, Target	Define the target, including the allocation mappings. The target is the destination to which the amounts that are defined by the source and basis are allocated.
Allocation Manager Rules - Residual	PF_AL_RESID_DEFN	EPM Foundation, Data Enrichment Tools, Allocation Manager, Rules, Residual	Assign residual amounts after the source amount has been allocated to dimensions in the target datamap. You can specify the residual account to use and other dimensions as applicable.
Rules - Offset	PF_AL_OFFSET_DEFN	EPM Foundation, Data Enrichment Tools, Allocation Manager, Rules, Offset	Specify offset target information based on dimensions in the target datamap.
Allocation Template	PF_AL_TMPLT_DEFN	EPM Foundation, Data Enrichment Tools, Allocation Manager, Allocation Template	Predefine a set of source, basis, and target datamaps that you can later associate with any allocation rule.

Determining the List of Dimensions to Use

Access the Fixed Dimension Setup page (EPM Foundation, Data Enrichment Tools, Allocation Manager, Fixed Basis Dimensions, Fixed Dimension Setup).



Fixed Dimension Setup page

Dimension Name Each dimension on this page is based on the FIX_BASIS datamap. This datamap is delivered with your system and consists of a generic table into which you can enter your organization-specific information. You can define a maximum of 12 dimensions for this datamap. PeopleSoft delivers this datamap with several dimensions; however, you can create more, if necessary.

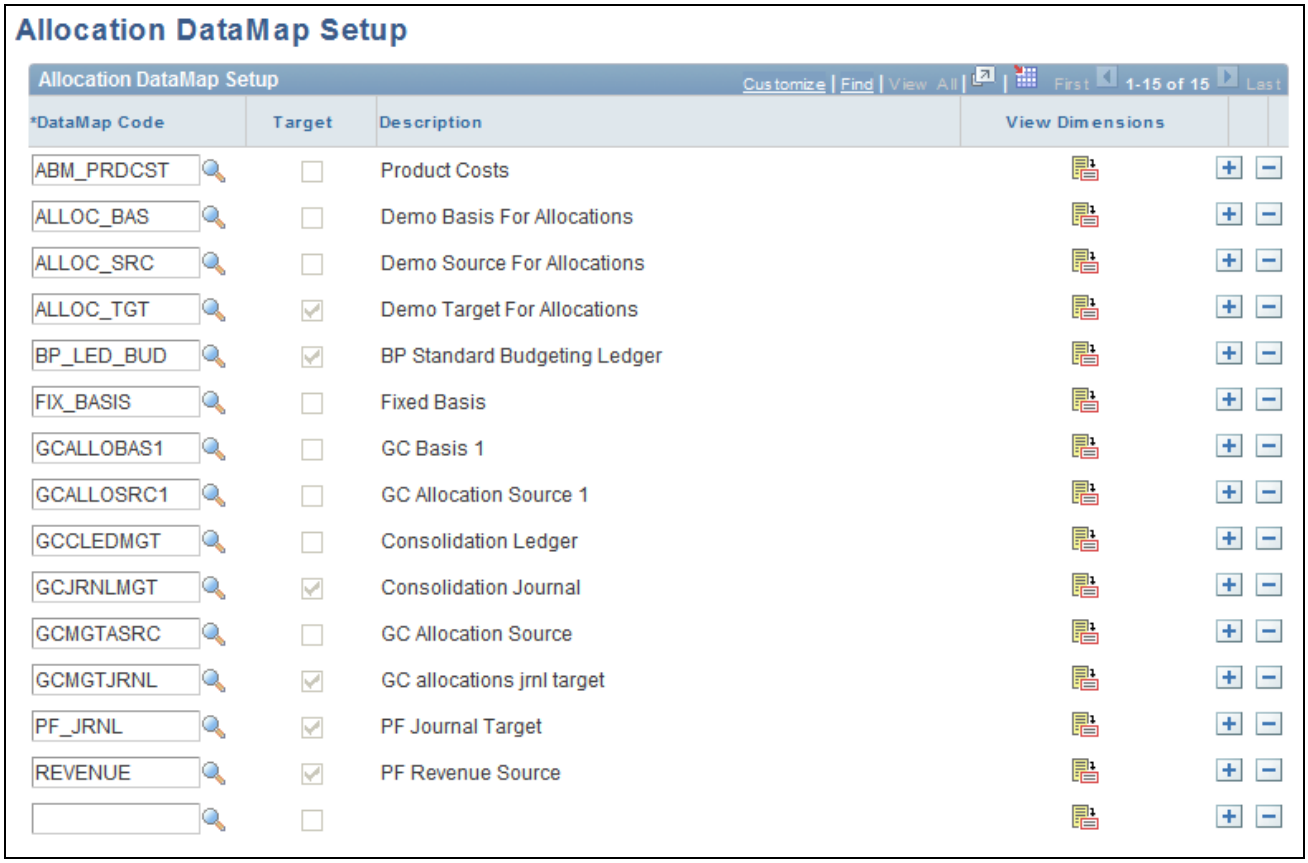
Lookup Table and **Lookup Tree** For each dimension, assign either prompt tables, trees, or both to select values.

Refresh Changes to the FIX_BASIS datamap do not automatically display on this page. Click the Refresh button to update the page with any additions or deletions to the datamap.

Note. You can only change dimensions. Do not change attributes or the measure.

Creating a List of Sources

Access the Allocation DataMap Setup page (EPM Foundation, Data Enrichment Tools, Allocation Manager, DataMap Setup, Allocation DataMap Setup).



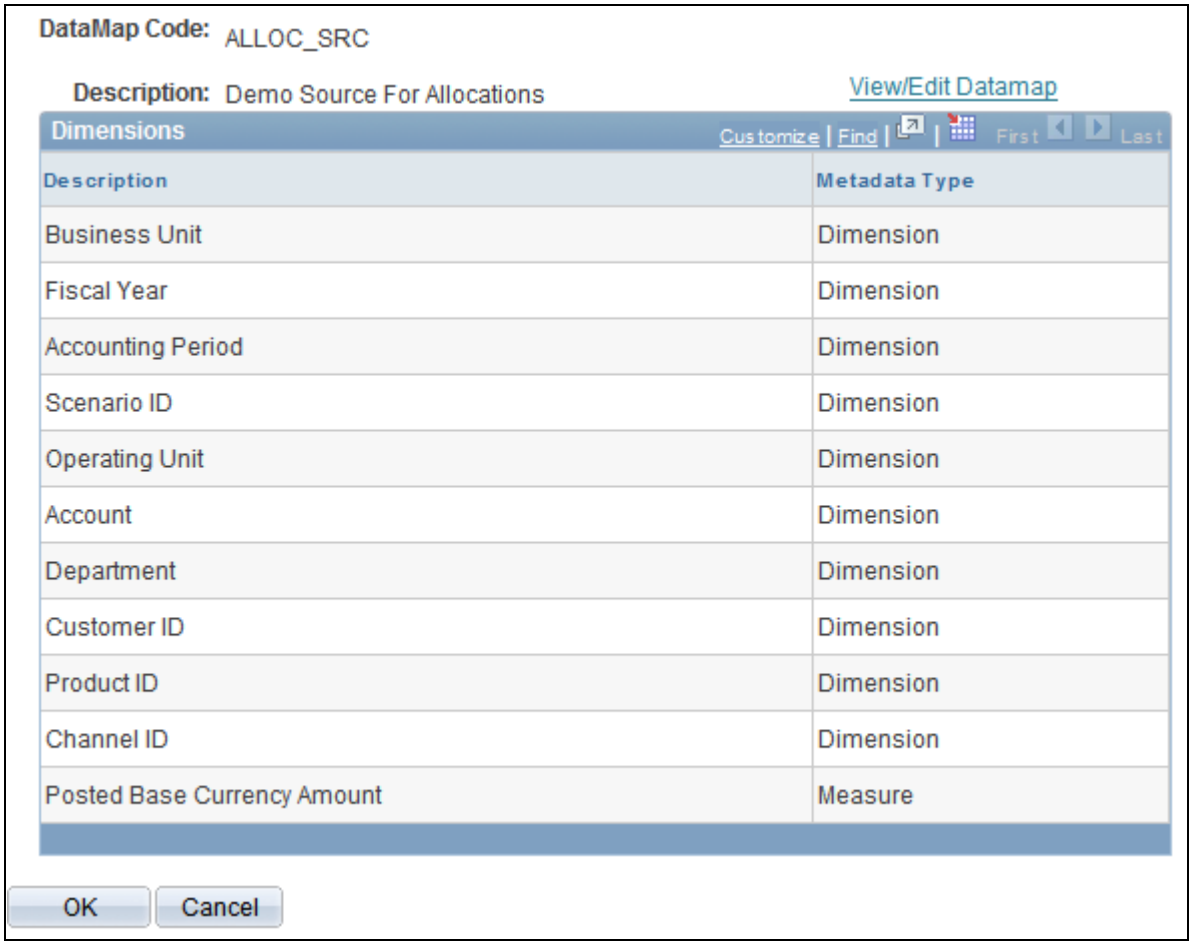
Allocation DataMap Setup page

DataMap Code This code identifies the datamap.

Note. You must create the datamaps before using the Datamap component.

Target If selected, the target indicates that the datamap can be used as a target. This check box is only an indicator. This is defined as part of the datamap definition.

Click the Select Dimensions button to access the Select Dimensions page and review the dimension fields in your datamaps. When you are done, click OK to return to the Allocation DataMap Setup page.



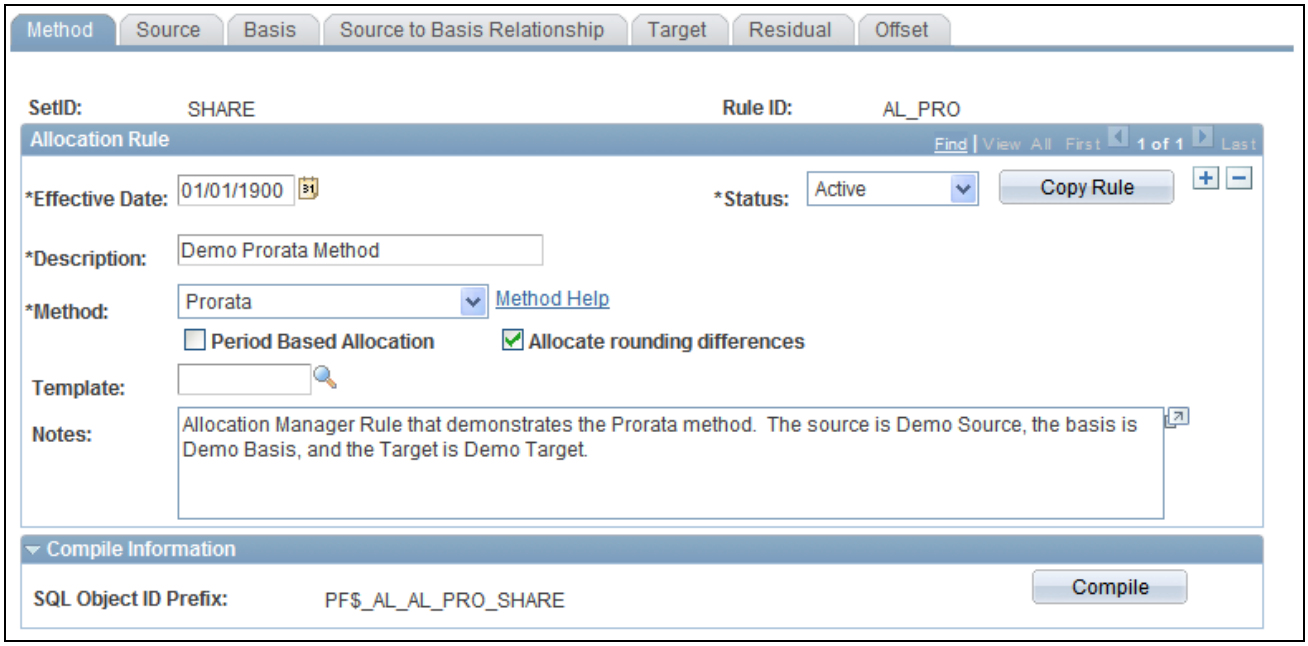
Select Dimensions page

See Also

Chapter 16, "Setting Up and Working with Metadata for the Operational Warehouse - Enriched," Setting Up Datamaps, page 363

Defining an Allocation Manager Rule

Access the Rules - Method page (EPM Foundation, Data Enrichment Tools, Allocation Manager, Rules, Method).



Method page

Method Select the calculation method for the allocation. The method determines the calculation options, such as how to determine the source amounts going to the targets.

Method options are: *Arithmetic*, *Copy*, *Prorata*, and *Spread Even.*

For the *Arithmetic* method, the Arithmetic Operation field becomes available for input.

For the *Prorata* method, the system prorates measures in source 1 across all common dimensions based on source 2 numbers.

For the *Spread Even* method, the system spreads source 1 measures evenly across all common dimensions based on source 2 numbers.

Method Help

Click this link to view additional information about the chosen calculation type.

For example, if you select the prorata method and click this link, the following information appears:

Divides the source amount proportionately among the targets based on basis measures.

The prorata method performs the following calculation:

*Source Measure * Basis Measure / Basis Total Source Measure = Sum(measures in source grouped by common dimensions and source mapped dimensions)*

Basis Measure = Sum(measures in basis grouped by common dimensions and basis mapped dimensions)

Basis Total = Sum(measures in basis grouped by common dimensions only)

Arithmetic Operation

For a method of *Arithmetic*, select the operation to use from *Addition (+)*, *Division (/)*, *Multiplication (*)*, and *Subtraction (-)*.

Period Based Allocation

Select this check box to specify period-based allocation.

This check box is only available when you select the *Prorata* and *Spread Even* methods.

Period-based allocation enables you to allocate from one period to one or more additional periods. For example, you may allocate quarterly source data into monthly target data based upon monthly basis data. To do this, a scenario ID must be specified for the basis on the Basis page. The scenario ID determines from which calendar to select the basis data. The calendar that is identified by the scenario that you selected on the Basis page determines which periods from the basis are used to perform the prorata or spread even processes.

Note. The source period must include one or more periods from the basis calendar. For example, if your source period is quarterly, your basis period should be monthly, weekly or daily.

Note. For period-based allocations, after the allocation is complete, the Merge application engine will move data directly from the temporary target table to the permanent target table.

Allocate Rounding Differences

Select this check box to enable rounding.

If selected the amount is rounded to 3 decimal points, or 83.333 per period.

In some allocation rules, the totals of the source amounts may not match the total amounts allocated. This may be due to the rule definition itself, or rounding errors introduced by the allocation. This feature will add the difference generated by the rounding to one row so that the total amount allocated is the same as the source amount.

You may also deselect this check box to disable rounding and increase system performance. If rounding is disabled, the system rounds all amounts that are prorated or distributed to the closest whole number for each period. This takes considerably less time than it would using rounding. For example, it is more time consuming to spread an amount of 1,0000 evenly across twelve periods because the results equals 83.33333333333 per period.

Template (optional) Select a template if you want to associate a set of predefined source, basis, and target DataMaps with the allocation.

Selecting a template will automatically populate the Source, Basis, and Target pages with the DataMaps specified in the template.

See Chapter 22, "Using Data Enrichment Tools," Defining an Allocation Template (Optional), page 592.

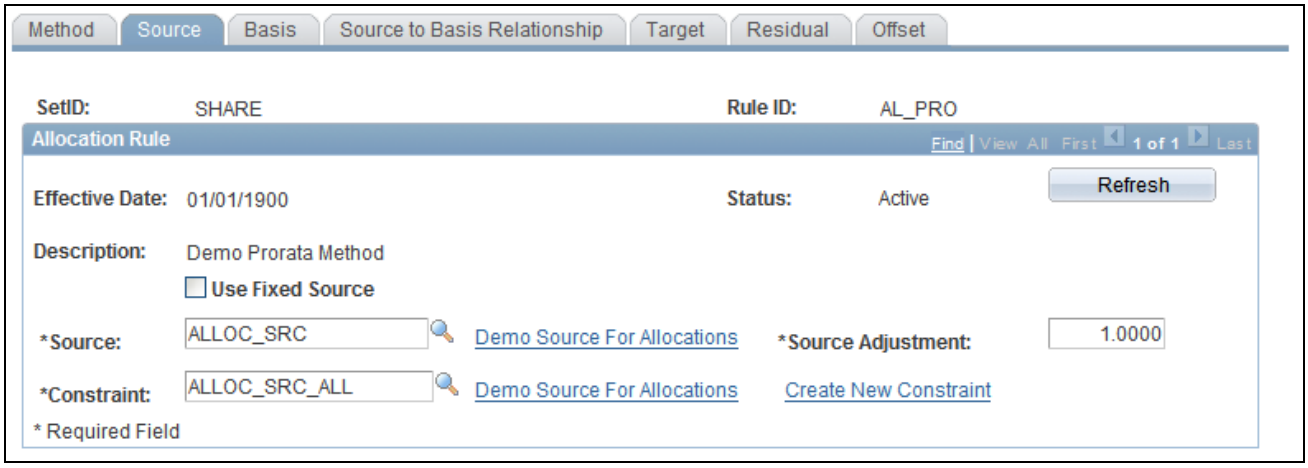
Copy Rule Click to access the Copy Rule page, on which you can enter a new rule ID and copy the rule information. Enter an effective date. The system default is today's date. Click OK to save the changes and return to the Method page.

Compile Click to individually compile the Allocation Manager rule. You can also use Mass Compile.

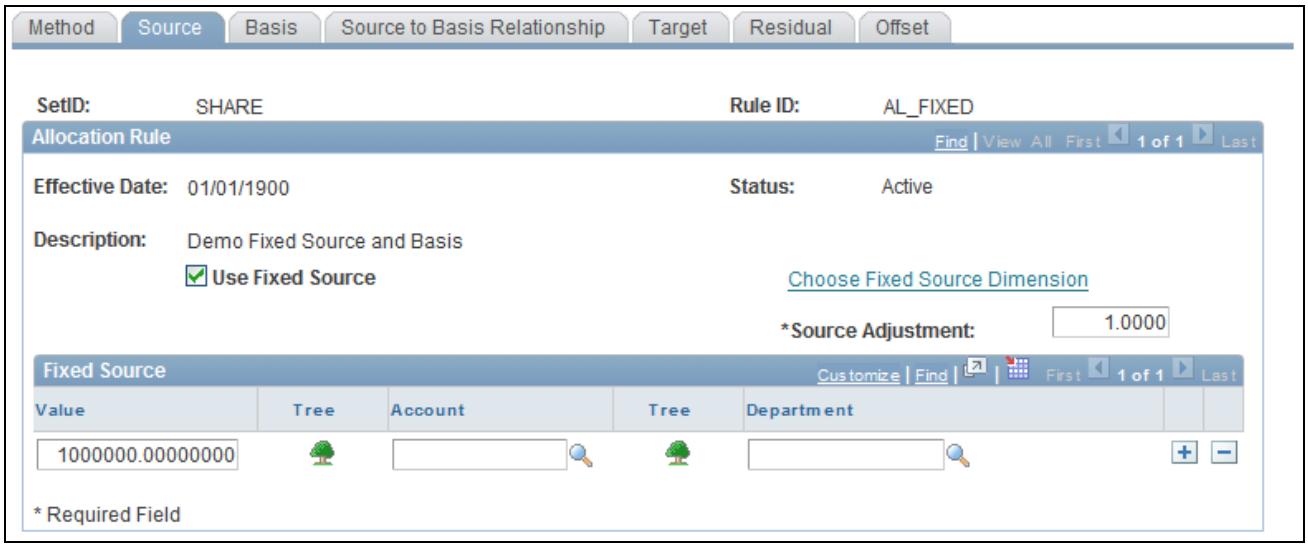
Note. You cannot use Mass Compile for allocation rules that are period-based or have the inter-business unit option selected. The system will skip these . You must compile these rules directly from the Allocation Manager Rules component.

Defining the Allocation Rule Source

Access the Rules - Source page (EPM Foundation, Data Enrichment Tools, Allocation Manager, Rules, Source).



Source page



Source page with fixed source option and Tree node button

Refresh In correction mode, click Refresh to update the page with any additions or deletions to the DataMaps.

Use Fixed Source Select this check box to enable the use of all available fixed dimensions.

See 'Using a Fixed Source' below.

Source Adjustment Enter a source adjustment against the source datamap.

This field defaults to 1, but you can input a larger number for a multiplying factor against source amounts.

You can also input a negative number adjustment so that the source amounts are multiplied by a negative number.

Selecting a Source

If you are not using fixed source, you must specify the source to use in the allocation.

Source Select a source from the available options. The options are based on the datamaps that are listed on the Allocation DataMap Setup page

Constraint Select a constraint for the source DataMap.

If the criteria that you selected requires meta-value objects, the Assign Values to Value Objects group box appears and you can assign values to the constraint at this time.

Demo Source for Allocations For the Source field, click the link to access the General Properties (DataMap) page and view the selected DataMap.

For the Constraint field, click the link to access the Constraint page and view the selected constraint.

Create New Constraint Click to access to the Constraint setup page and create a new constraint.

Using a Fixed Source

Choose Fixed Source Dimension For a fixed source, click this link to access the Fixed Source Dimension Setup page and select your dimensions.

Click OK to return to the Source page.

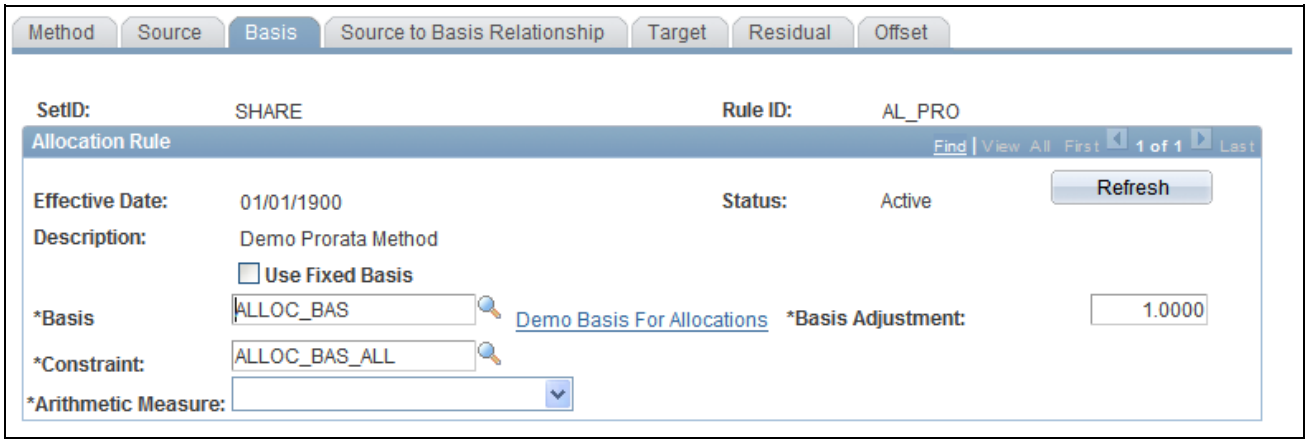
Fixed Source Value If you select a fixed source, populate the value field for each dimension that you select

You must assign a value for each fixed dimension.

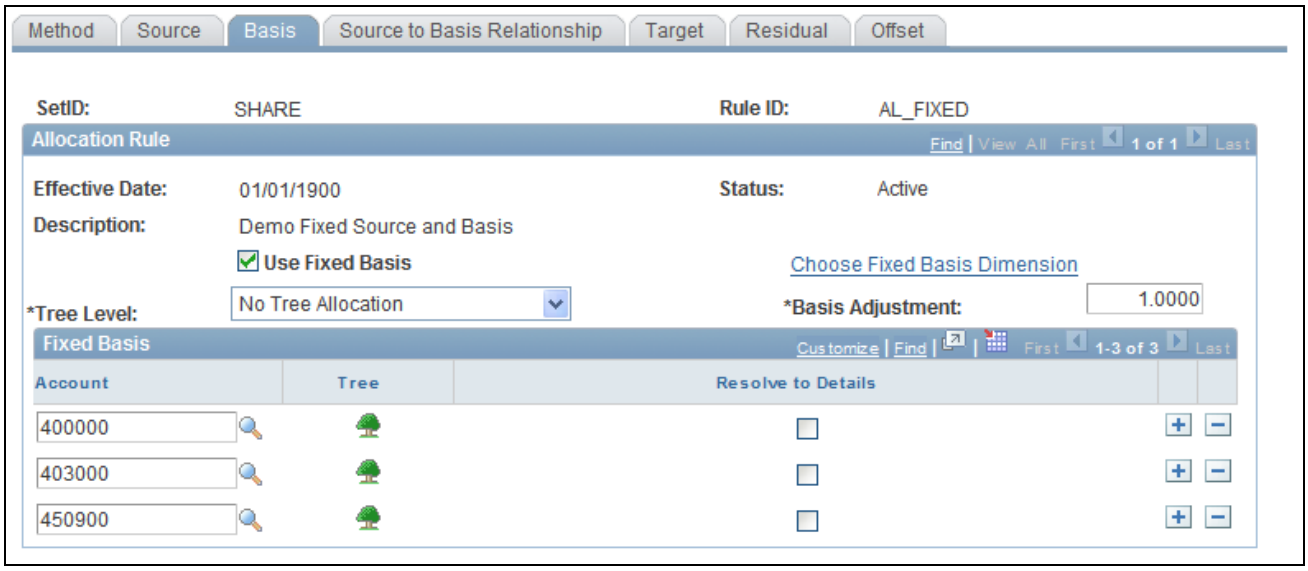
Where applicable, click the Tree Node button alongside the Fixed Source Value field to access the Tree Viewer page. Navigate through the tree by collapsing and expanding the tree. When you have located the tree node you want to use, click the Select button to select this node and return to the Source page.

Defining the Allocation Rule Basis

Access the Rules - Basis page (EPM Foundation, Data Enrichment Tools, Allocation Manager, Rules, Basis).



Basis page



Basis page with fixed basis option

Refresh In correction mode, click Refresh to update the page with any additions or deletions to the DataMaps.

Use Fixed Basis Select to use a fixed basis for this rule.

Basis Adjustment Enter a basis adjustment against the basis datamap.

This field defaults to 1, but you can input a larger number for a multiplying factor against basis amounts.

You can also input a negative number adjustment so that the basis amounts are multiplied by a negative number.

Note. The allocation application engine applies the basis adjustment to the basis before it is used within the allocation calculation.

Note. If you selected the copy method, only the Basis Adjustment field is available on this page.

Selecting a Basis

Basis Select a basis from the available options. The options are based on the datamaps that are listed on the Allocation DataMap Setup page

Constraint Select a constraint for the basis DataMap.

If the criteria that you selected requires meta-value objects, the Assign Values to Value Objects group box appears and you can assign values to the constraint at this time.

Demo Basis for Allocations

For the Basis field, click the link to access the General Properties (DataMap) page and view the selected DataMap.

For the Constraint field, click the link to access the Constraint page and view the selected constraint.

Create New Constraint

Click to access to the Constraint setup page and create a new constraint.

Base Scenario

If you selected the Period Based Allocation check box on the Method page, select a basis scenario. Allocation Manager uses the calendar for the basis scenario that you selected to determine from which periods to use to prorate or spread the allocations. The target has the same fiscal year and accounting period as the basis.

Arithmetic Measure

For the arithmetic method, select a measure. The available measures are based on the datamap that you selected as the basis.

Using a Fixed Basis

Choose Fixed Basis Dimension

For a fixed basis, click this link to access the Fixed Basis Dimension Setup page and select your dimensions.

Click OK to return to the Basis page.

Prorata Measure

For the prorata method, select a prorata measure.

The available measures are based on the datamap that you selected as the basis.

Tree Level

Select the tree level to which you want to resolve your allocation. Your options are: *No Tree Allocation, Allocate to leaf level,* or *Allocate to next node.*

Select the values for each of the dimensions that you selected on the Fixed Basis Dimension Setup page.

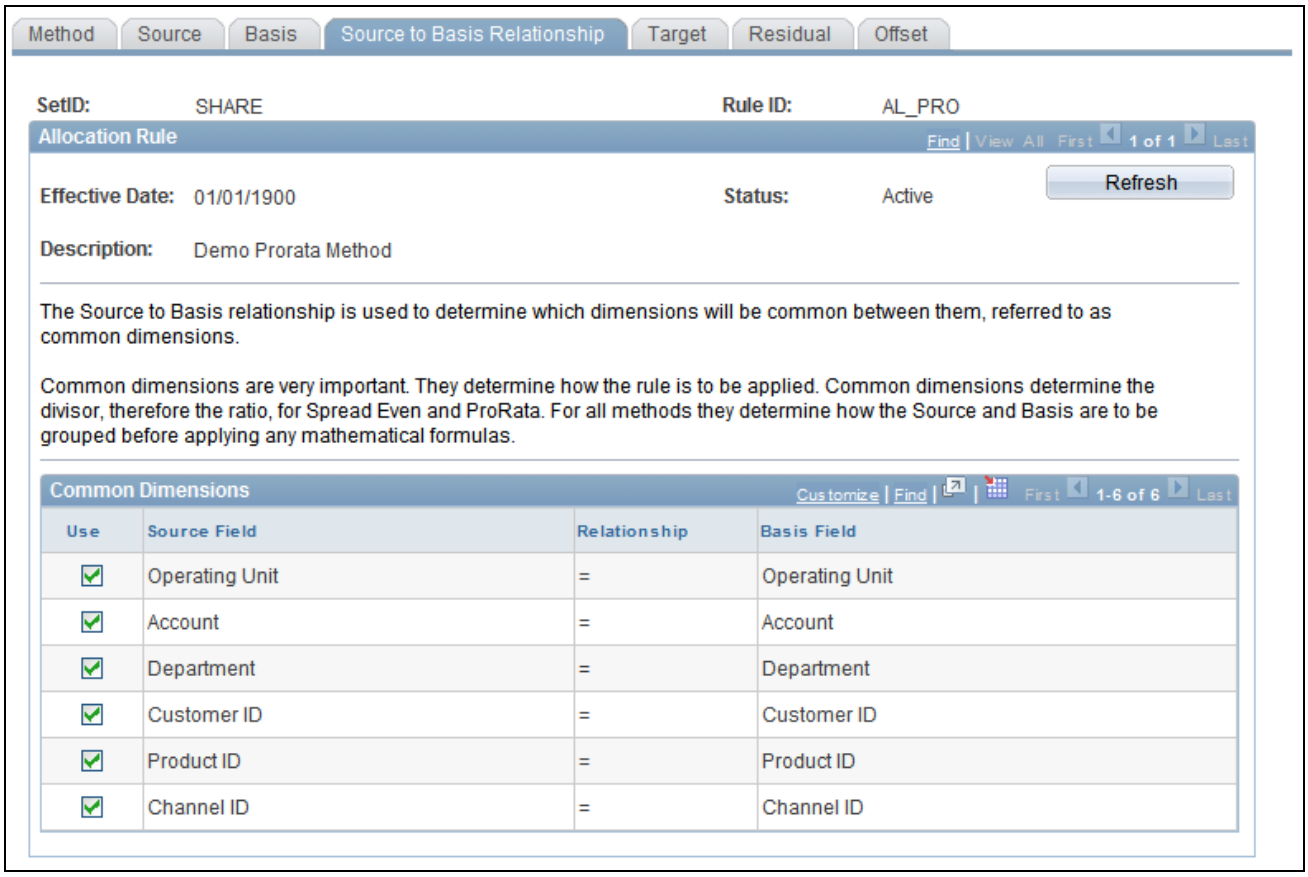
Click the Use Tree Node next to any dimension field in which it displays to select the value using the Tree Viewer.

Resolve to Details

For tree node values, select this check box to sort your tree node information in ascending order.

Reviewing Common Dimensions

Access the Rules - Source to Basis Relationship page (EPM Foundation, Data Enrichment Tools, Allocation Manager, Rules, Source to Basis Relationship).



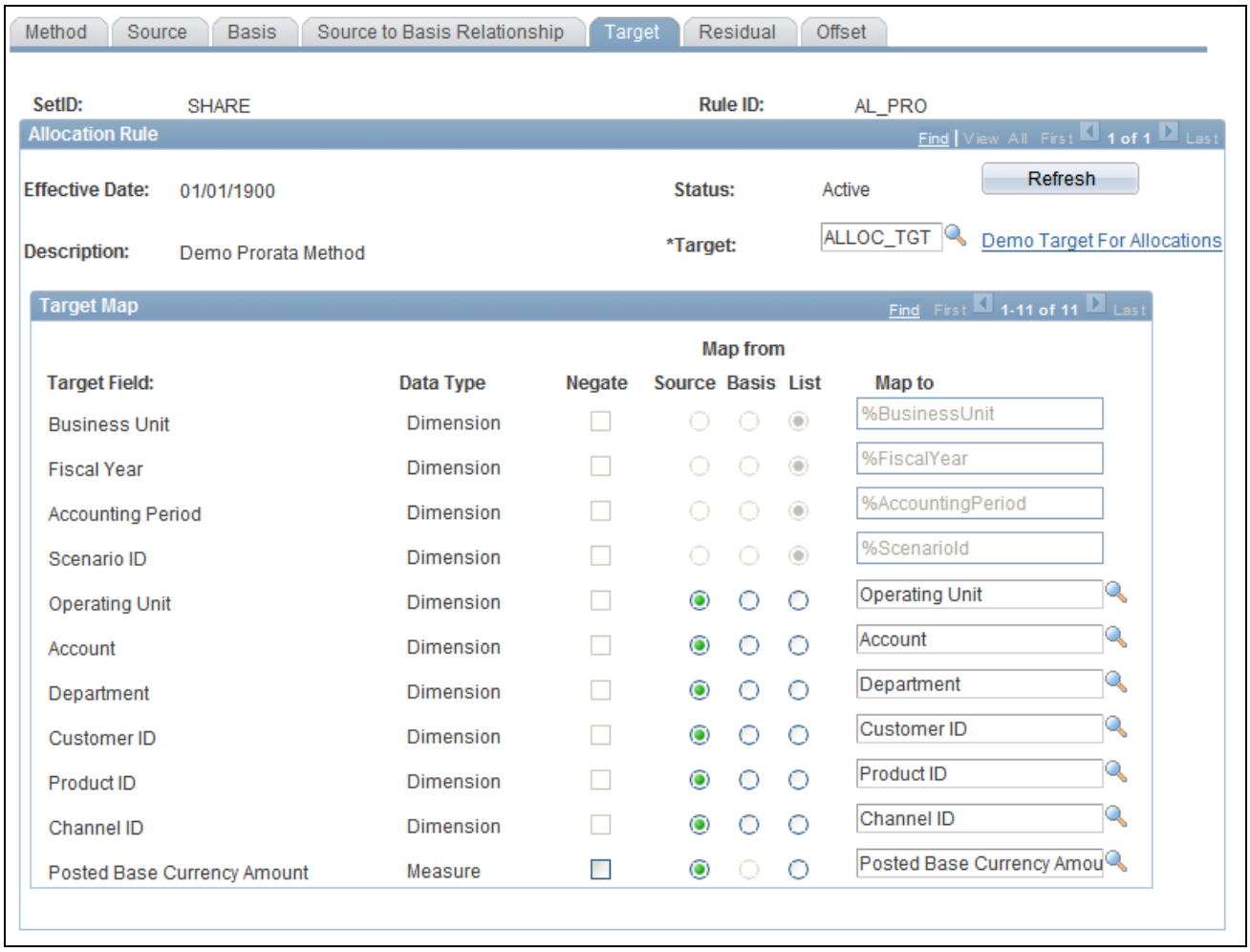
Source to Basis Relationship page

This page lists the common dimensions. Common dimensions are very important as they are used to determine the divisor, therefore the ratio, for the spread even and prorata methods. For all methods, they determine how the source and basis are to be grouped before applying any mathematical formulas. Select from which source to base dimensions by selecting the Use check box.

The system determines the common dimensions using the descriptions of the datamaps for the source and basis. It considers that any source and basis fields with the same description represent the same information and are therefore a common dimension if the fields are marked as dimensions on both the source and basis. When looking for common dimensions, the system ignores case and spaces at the beginning and end of the description.

Defining the Target

Access the Rules - Target page (EPM Foundation, Data Enrichment Tools, Allocation Manager, Rules, Target).



Target page

Target Select the target datamap for the allocation.

Demo Target for Allocations Click the link to access the General Properties (DataMap) page and view the selected target DataMap.

Target Field and **Data Type** The system populates these fields based on the target you select.

Negate Select the Negate check box to reverse the sign of the amount when the data is copied to the target table.

The Negate check box applies only to target fields with a data type of *Measure*.

(Map from) Source Select to use the field values from the source records.

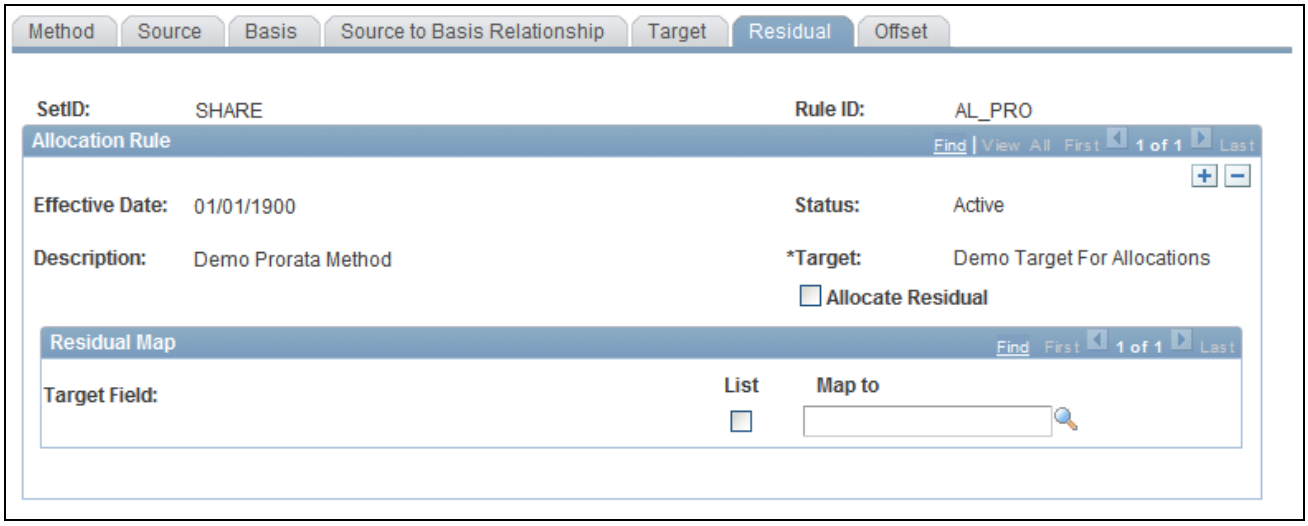
(Map from) Basis Select to use the field values from the basis records.

(Map from) List Select to use predefined value objects.

Map To Specify the mapping for your selection.

Assigning Residual Amounts (Optional)

Access the Rules - Residual page (EPM Foundation, Data Enrichment Tools, Allocation Manager, Rules, Residual).



Residual page

Allocate Residual Select this check box to assign residual amounts after the source amount has been allocated to dimensions in the target datamap.

The page displays the target fields for the residual map.

List Select this check box to allocate residual amounts of a target field to a value object.

The Map To field displays the value object.

Map To If you select the List check box, the system displays the value object. Otherwise, you can enter the appropriate dimension value to use in this field. For example, to allocate residual amounts to a specific account, do not select the List check box. Simply select an account from the list of departments. You must select a value for at least one of the dimensions.

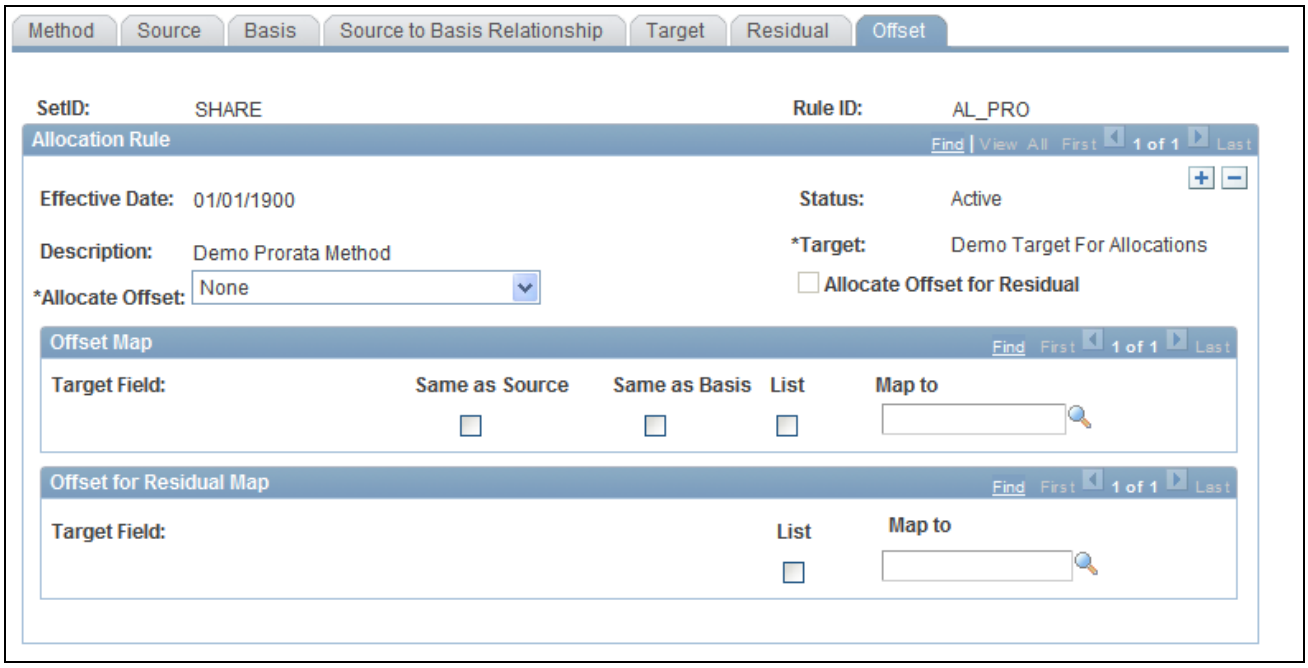
The following table provides an example of an arithmetic allocation rule in which the residual source amount is set up with a target department value of RES (Residual). DIV1, 2, and 3 use a total of 400. The residual amount is 600, so that amount is placed in the RES account.

Source		***Basis***		***Target***	
Source Department	Amount	Basis Department	Unit	Department	Amount
CORP	1000	DIV1	0.1	DIV1	100
		DIV2	0.1	DIV2	100

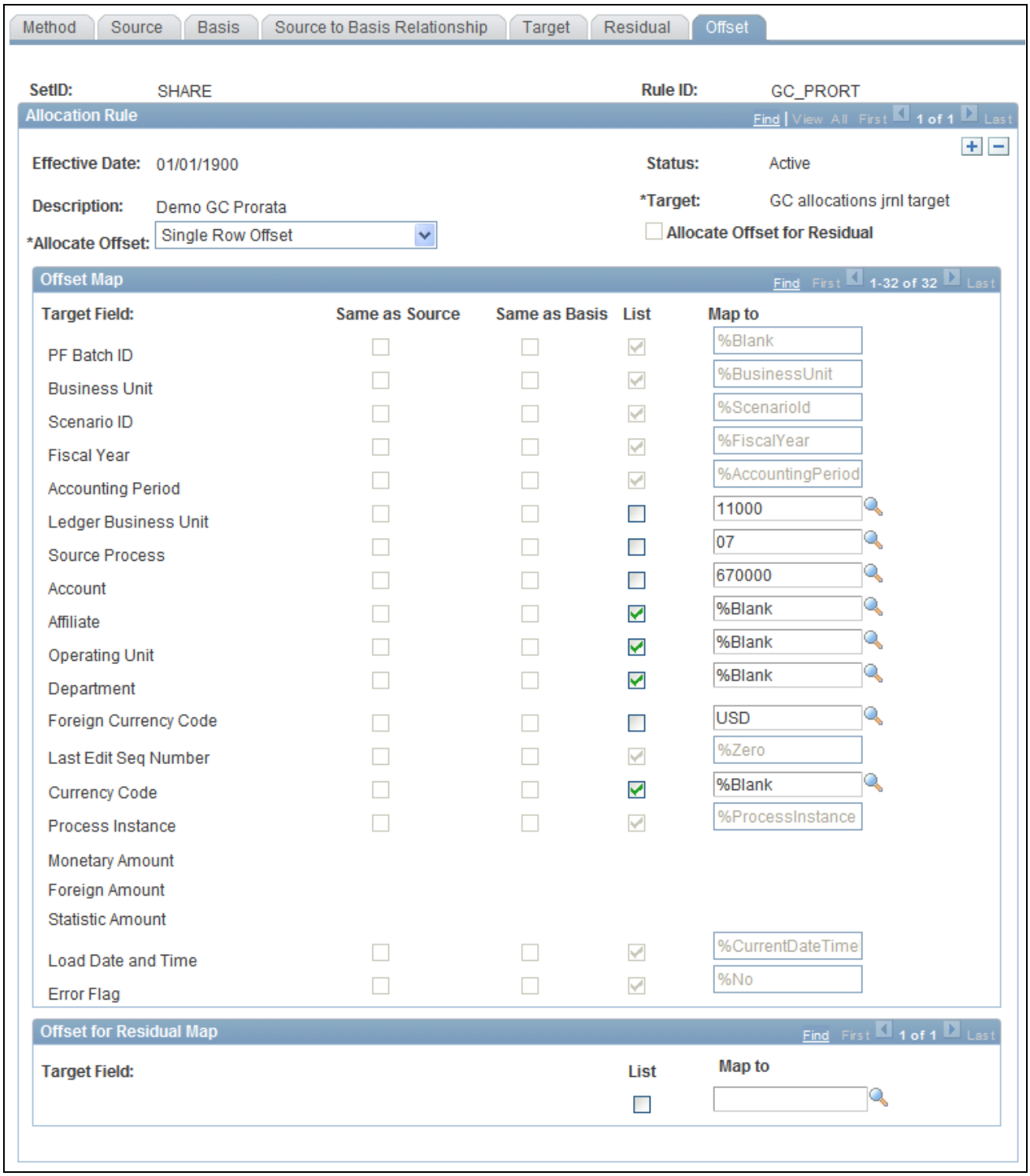
Source		*Basis*		*Target*	
		DIV3	0.2	DIV3	200
				RES (Residual Account)	600

Specifying Offset Target Information (Optional)

Access the Rules - Offset page (EPM Foundation, Data Enrichment Tools, Allocation Manager, Rules, Offset).



Offset page



Offset page with single row offset option

Allocate Offset Select this check box to allocate offset target information based on dimensions in the target datamap.

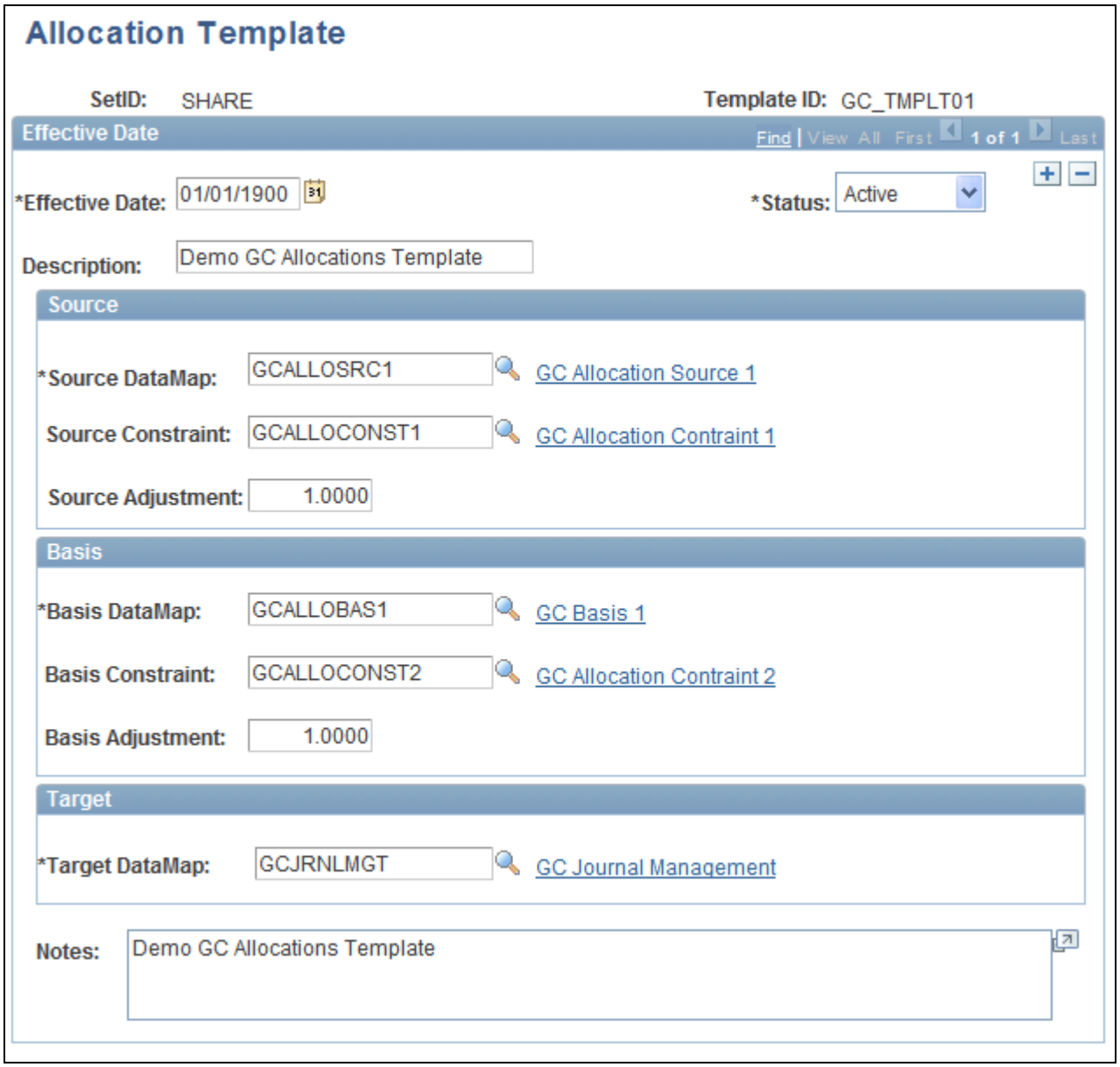
The system displays the target fields for the offset map.

Allocate Offset for Residual	Select this check box to allocate offset amounts for a residual amount. The system displays the target fields for the offset for residual map.
Same as Source	Instead of selecting a dimension value for the offset, select this check box to use the source dimension value you already specified.
Same as Basis	Instead of selecting a dimension value for the offset, select this check box to use the basis dimension value you already specified.
List	Select this check box to allocate the offset to a value object. The Map To field displays the value object.
Map To	If the List check box is selected, this field displays the value object and is not available for input. If the List check box is not selected, you can enter the appropriate dimension value to use in this field. For example, to allocate offset amounts to a specific account, do not select the List check box. Simply select an account from the list of accounts.

Note. You must select a value for at least one of the dimensions.

Defining an Allocation Template (Optional)

Access the Allocation Template page (EPM Foundation, Data Enrichment Tools, Allocation Manager, Allocation Template).



Allocation Template page

Use the Allocation Template page to predefine a set of source, basis, and target DataMaps that you can later associate with any allocation rule.

Fields Common to this Page

Create New DataMap Click the link to access the General Properties (DataMap) page and either create a new DataMap or select an existing DataMap.

Note. Once you select a DataMap and save the template, the link name changes to the DataMap description provided on the General Properties (DataMap) page. Also, the link will now access the General Properties (DataMap) page specific to the *selected* DataMap, where you can view or update details about the DataMap

Create New Constraint Click this link to access the Constraints page and view details about the selected constraint.

Click the link to access the Constraints page and either create a new constraint or select an existing constraint.

Note. Once you select a constraint and save the template, the link name changes to the constraint description provided on the Constraint page. Also, the link will now access the Constraint page specific to the *selected* constraint, where you can view or update details about the constraint.

Source

Source DataMap Select a source DataMap for the allocation template.

You can only select from DataMaps that are defined on the Allocation DataMap Setup page.

Source Constraint Select a source DataMap constraint for the allocation template.

You can only select from constraints that are defined for this DataMap on the Constraints page.

Source Adjustment Enter a source adjustment against the source datamap.

This field defaults to 1, but you can input a larger number for a multiplying factor against source amounts.

You can also input a negative number adjustment so that the source amounts are multiplied by a negative number.

Basis

Basis DataMap Select a basis DataMap for the allocation template.

You can only select from DataMaps that are defined on the Allocation DataMap Setup page.

Basis Constraint Select a basis DataMap constraint for the allocation template.

You can only select from constraints that are defined for this DataMap on the Constraints page.

Basis Adjustment Enter a basis adjustment against the basis datamap.

This field defaults to 1, but you can input a larger number for a multiplying factor against basis amounts.

You can also input a negative number adjustment so that the basis amounts are multiplied by a negative number.

Note. The allocation application engine applies the basis adjustment to the basis before it is used within the allocation calculation.

Target

Target DataMap Select a target DataMap for the allocation template.

You can only select from DataMaps that are defined on the Allocation DataMap Setup page.

Defining Allocation Manager Rule Sets

This section provides an overview of Allocation Manager rule sets and discusses how to

- Set up Allocation Manager rule sets.
- Define post-processing routines for an allocation rule set.

Understanding Allocation Manager Rule Sets

You must define an Allocation Manager rule set for any Allocation Manager rules that you want to process. You can also use rule sets to combine multiple rules in one run of the Allocation Manager engine. Rules in a rule set run in the order that is shown on the Allocation Manager Rule Sets page. The first rule must complete successfully before the second rule starts. The system places the data from the first rule in a temporary table (which is assigned by the engine) that will be the source of the second rule, and so on.

Note. A rule must be included in a rule set, even if only one rule is processed.

After you define a rule set, you can create post-processing routines that further process the results of the rule set. You can also tie the routines to a particular rule set so that they automatically run when the rule set completes. The following sections summarize specific features of the post-processing routines.

Auditing Allocation Rules

This feature provides visibility into the inputs and rules that calculate your allocations. Runtime information, source data transformations, and basis data transformations are some of the data that is captured by the audit feature. The data is captured in logs and is stored in the corresponding Operational Warehouse - Enriched (OWE) tables. The following table contains a complete list of delivered audit logs and the data warehouse table in which the data reside:

Log Type	*Corresponding OWE Table*	*Data Captured*
Runtime Log	PF_AL_AUD_RUNLG	Captures all rule and runtime values. All the following table entries are tied together by this unique runtime log key.
Common Dimensions Log	PF_AL_AUD_CDIM	Captures common dimensions.
Source & Basis to Target Mapping Log	PF_AL_AUD_MAP	Mapping from source/basis columns to target column.

Log Type	*Corresponding OWE Table*	*Data Captured*
Source Log	PF_AL_AUD_SRC	Captures input source data.
Basis Log	PF_AL_AUD_BAS	Captures input basis data.
Target Log	PF_AL_AUD_TGT	Captures target data.
Mapping Logs	PF_AL_AUD_SRMAP PF_AL_AUD_BSMAP PF_AL_AUD_TGMAP	Captures source, basis, and target name mappings.

You can view the audit logs using any SQL Query Tool.

Note. If a Ledger ID is not entered in the post processing page, a batch ID cannot be generated for the audit trail.

Merging Allocation Results to Permanent Tables

After an allocation rule set completes processing, the results are stored in a temporary target data warehouse table. This feature automatically loads the results that are stored in the temporary target table to a permanent non-journal target table—for example, PS_PF_AL_TGT.

This feature should be used if the target is not a journal table. The target tables should have the *Merge Allowed* check box selected on the Record Metadata page.

Note. Merging occurs automatically for period-based allocations and post processing is not required.

Validating Journals

If you load your data to a journal target table, this feature ensures the journal dimensions have valid values. For example, you might create allocation rules at the beginning of the year. At the time of allocation rule creation, you specified account 100 as one of the target accounts for the allocation. Account 100 is a valid account. Suppose that during the year, account 100 is deactivated. If the allocation process is run and validate journal is checked, the process creates an error, as account 100 is no longer valid. Invalid journal dimensions are sent to the journal error table (PF_JRNL_E00).

Posting Journals

This feature calls the PF_POST application engine to the post a journal entry to the ledger indicated by the Ledger ID.

Reversing Journals

If you load your data to a journal target table, this feature automatically reverses the journal entry. This process creates an identical reversed set of journal entries and results in two journal entries: the original journal and another with all the amount signs reversed. In addition, you can specify whether you want to reverse a journal for the same or alternate accounting period.

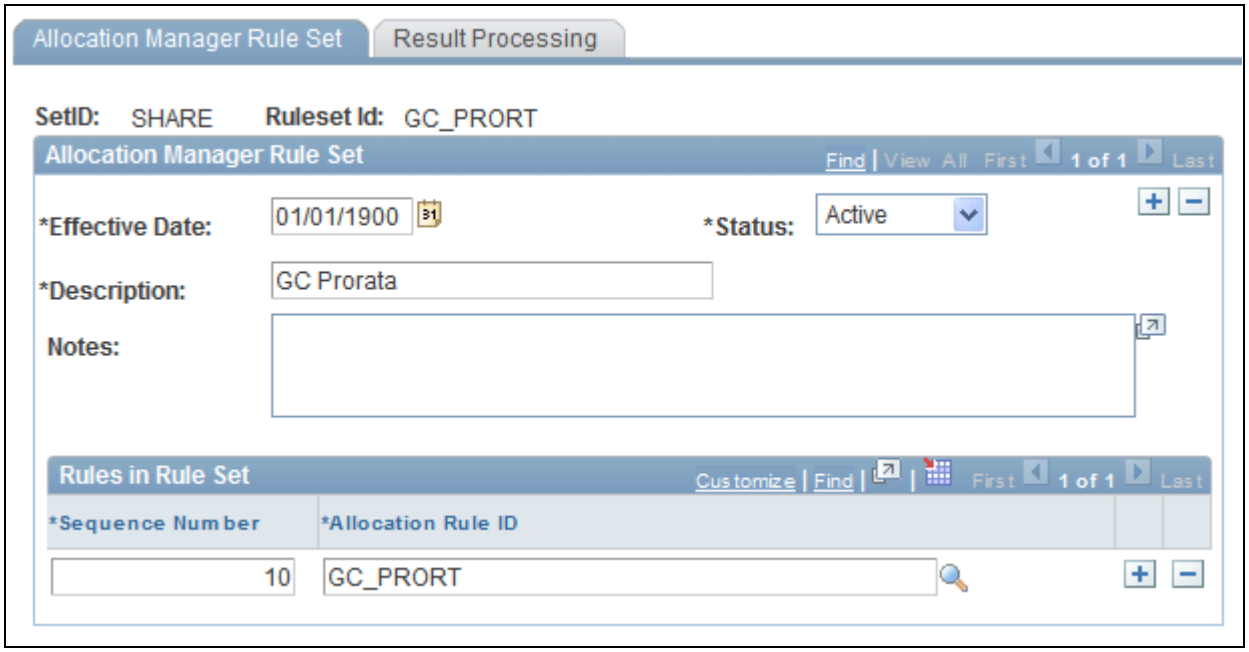
If you select the validate and post options with the reverse journal option, the reversed journal entries will also be validated and posted. In the case of specific period, proper error checking is done by the Allocation Manager to determine if the entered fiscal year and accounting period are valid.

Pages Used to Define Allocation Manager Rule Sets and Post-Processing Routines

Page Name	*Definition Name*	*Navigation*	*Usage*
Allocation Manager Rule Set	PF_AL_RULESET_DEFN	EPM Foundation, Data Enrichment Tools, Allocation Manager, Rule Set, Allocation Manager Rule Set	Set up an Allocation Manager rule set by defining the rules to include in the rule set.
Result Processing	PF_AL_PROC_OPTIONS	EPM Foundation, Data Enrichment Tools, Allocation Manager, Rule Set, Result Processing	Define post-processing routines that further process the results of the rule set.

Setting Up Allocation Manager Rule Sets

Access the Allocation Manager Rule Set page (EPM Foundation, Data Enrichment Tools, Allocation Manager, Rule Set, Allocation Manager Rule Set).



Allocation Manager Rule Set page

Ruleset ID Displays the unique identifier for this rule set definition.

Rules in Rule Set

Sequence Number Enter a number for the first rule ID in the rule set (for example 100).

This number represents the sequencing order in which the rules run.

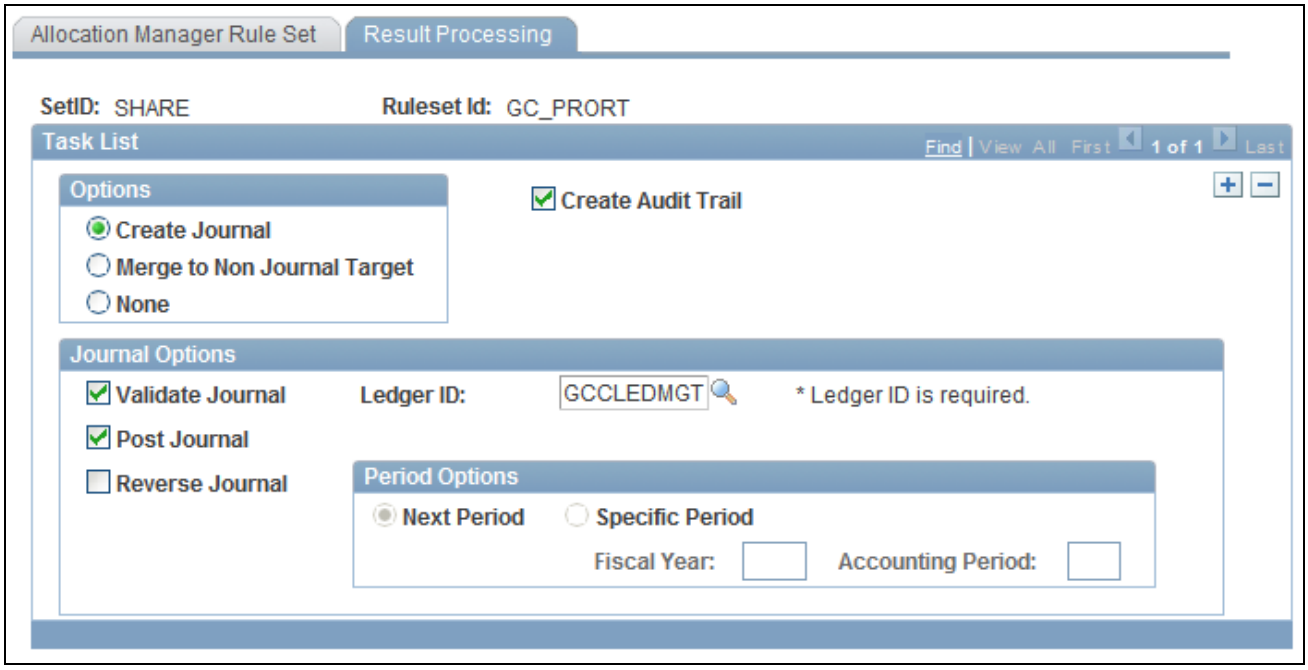
The next rule ID to run might have a sequence number of 200. The actual sequence number is not important; it represents the order in which you want to process rules. The sequence must be unique.

Allocation Rule ID Select the Allocation Manager rule to include in the rule set.

You create the rules using the Allocation Manager Rules component.

Defining Post-Processing Routines for an Allocation Rule Set

Access the Results Processing page (EPM Foundation, Data Enrichment Tools, Allocation Manager, Rule Set, Result Processing).



Result Processing page

Ruleset ID Displays the unique identifier for this rule set definition.

Create Audit Trail Select this check box to generate audit logs of the inputs and rules that calculate your allocations.

Runtime information, source data transformations, and basis data transformations are some of the data that is captured by the audit logs. You can view the audit logs using PeopleSoft Query.

Options

Create Journal Select this option to automatically load the allocation output into a target Journal line table (for example, PS_PF_JRNL_F00).

At runtime, the PF_EDIT process is called and a batch header (PF_BATCH_TBL) is created, which moves the journal lines from a temporary table to a permanent journal table. You can view results of this process using any SQL query tool such as Toad, Query Analyzer, and so on. When this option is selected, the Journal Options group box becomes available.

Merge to Non Journal Target Select this option to automatically load the allocation output into a non-journal permanent target table. At runtime, the PF_MERGE process is called and loads the allocation output from the temporary table to a permanent table that you specify in the target setup page. You can view results of this process using any SQL query tool such as Toad, Query Analyzer, and so on.

None Select this option if you do not want to load the allocation output into a permanent target table. When this option is selected, data from the allocation process remains in the temporary table. This option is the default for the upgrade process.

Journal Options

Validate Journal Select this check box to have the PF_EDIT engine validate the journal dimensions that are specified during the rule setup.

When this option is selected, the system validates all journal dimensions. The system also rejects any values that do not appear on the corresponding prompt table. Invalid values are sent to the journal error table (PF_JRNL_E00).

If you do not select this option, the journal is created without validation and will be posted regardless of errors. It is recommended that you select this option. Note, however, that the validation process does increase processing time.

Ledger ID Select the ledger ID to generate a corresponding batch ID, which is used to post the journal to the ledger.

Post Journal Select this check box to store the data in a journal table and automatically post it to a permanent ledger table (for example, PF_LEDGER_F00). At runtime, the system calls PF_POST, which updates the ledger.

Reverse Journal Select this check box to reverse a journal entry for the same or alternate accounting period. This process results in two journal entries: the original journal and another with all the amount signs reversed. When this option is selected, the Reverse Journal Options group box becomes available and enables you to further specify a particular accounting period for the reversed journal.

Next Period Select this option to reverse a journal for the next accounting period.

Specific Period Select this option to reverse a journal for a specific accounting period.

Fiscal Year Enter the fiscal year for which you want to reverse the journal.

Accounting Period Enter the accounting period for which you want to reverse the journal.

The following table provides examples of the edit, validate, and post options used in combination with one another and the resulting output of the combination:

Options Selected	*Result*
Edit in Journal and Validate Journal	Allocation output is automatically loaded into a target journal line table. A journal with dimensions is checked for balancing and batch header with a valid status.
Edit in Journal and Post Journal	The allocation output is automatically loaded into a target journal line table. The post process (PF_POST) is called to post data to a specified ledger.
Edit in Journal, Validate Journal, and Post Journal	The allocation output is automatically loaded into a target journal line table. A journal with dimensions is checked for balancing and batch header with a valid status. The post process (PF_POST) is called to post data to a specified ledger.

Please note that these are merely examples of edit, validate, and post option-combinations; you can choose a number of different combinations.

Renaming Allocation Manager Rules

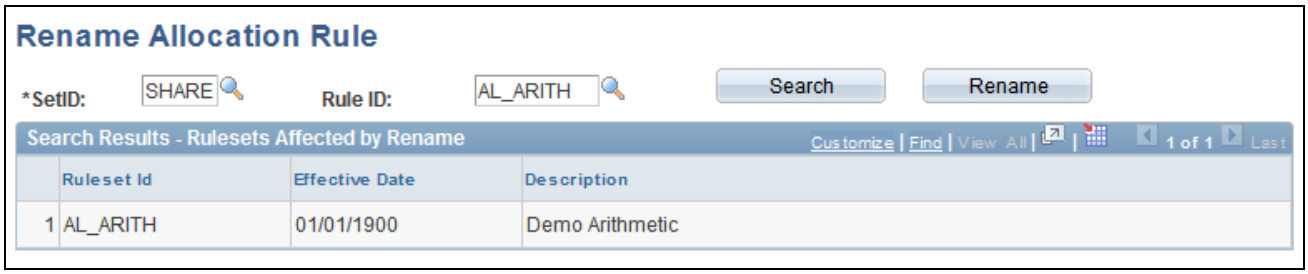
This section discusses how to rename an Allocation Manager rule.

Page Used to Rename Allocation Manager Rules

Page Name	*Definition Name*	*Navigation*	*Usage*
Rename Allocation Rule	PF_AL_RULE_RENAME	EPM Foundation, Data Enrichment Tools, Allocation Manager, Rename Rules, Rename Allocation Rule	Rename existing Allocation Manager rules.

Renaming an Allocation Manager Rule

Access the Rename Allocation Rule page (EPM Foundation, Data Enrichment Tools, Allocation Manager, Rename Rules, Rename Allocation Rule).



Rename Allocation Rule page

SetID and **Rule ID** Select the setID and rule to rename.

Search Click the Search button. The system populates the Search Results grid with a list of the rule sets that affected by the renaming process.

Rename To proceed with the renaming process, click the Rename button and enter a new name for the rule. Every instance of the original rule name is changed to the new name.

Associating Allocation Manager Rule Sets With a Job and Processing Allocations

The next steps required to process allocations include associating an allocation rule set with a job and running the Allocation Manager engine.

Each unique run of the Allocation Manager engine is assigned a job ID so you must associate the rules with the actual job that will run them using the Allocation Manager Rule Set Job Association page. When the jobstream that contains the defined Allocation Manager job ID runs, it uses the setup on the Allocation Manager Rule Set Job Association page to find the Allocation Manager rule set to execute.

Note. Each time that you run an EPM engine, it must have a unique job ID. These jobs must then be run in a jobstream.

Prerequisites

Before you can associate rule sets with job, you must:

- Set up the Allocation Manager job ID.
- Set up a jobstream for the job ID.

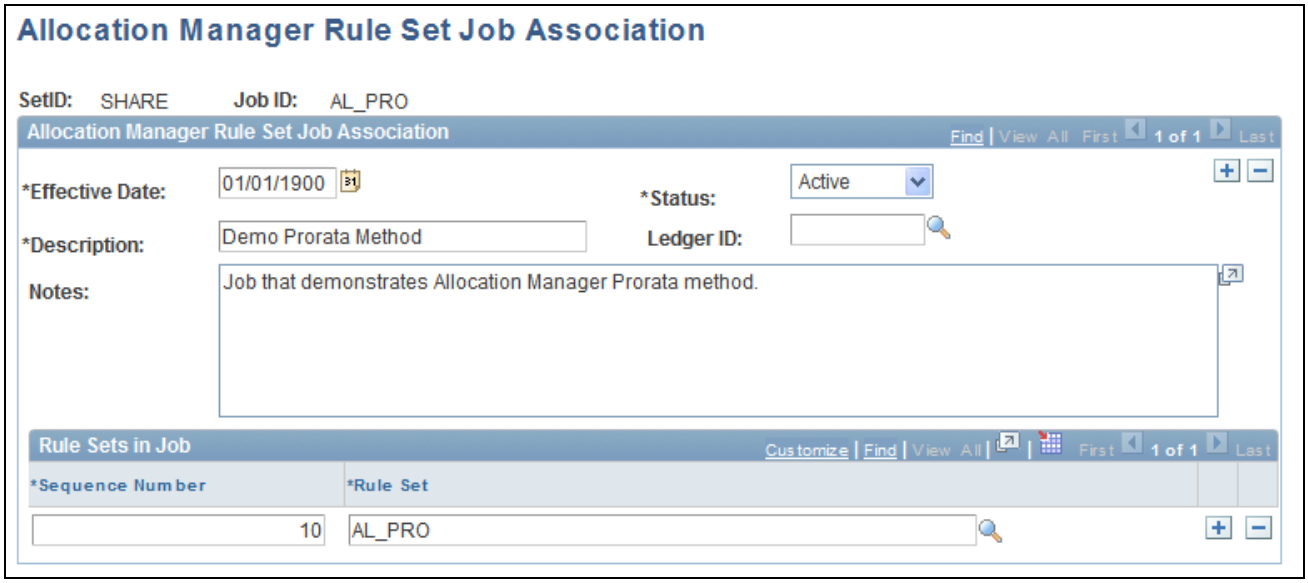
See Chapter 20, "Streamlining Processing with Jobstreams," page 461.

Pages Used to Associate Allocation Manager Rule Sets with a Job and Process Allocations

Page Name	*Definition Name*	*Navigation*	*Usage*
Allocation Manager Rule Set Job Association	PF_AL_JOB_RSET_DEF	EPM Foundation, Data Enrichment Tools, Allocation Manager, Job Association, Allocation Manager Rule Set Job Association	Define rule sets to be run with a given job and the order in which these rule sets are run.
Run Allocations	PF_RUN_ALLOCATION	EPM Foundation, Data Enrichment Tools, Allocation Manager, Run Allocations, Run Allocations	Process allocation rules.

Defining Allocation Manager Rule Sets to Run for a Given Job

Access the Allocation Manager Rule Set Job Association page (EPM Foundation, Data Enrichment Tools, Allocation Manager, Job Association, Allocation Manager Rule Set Job Association).



Allocation Manager Rule Set Job Association page

Job ID Displays the unique instance of an Allocation Manager engine in a jobstream.

Ledger ID Select a ledger ID. This is required for the Profit Manager.

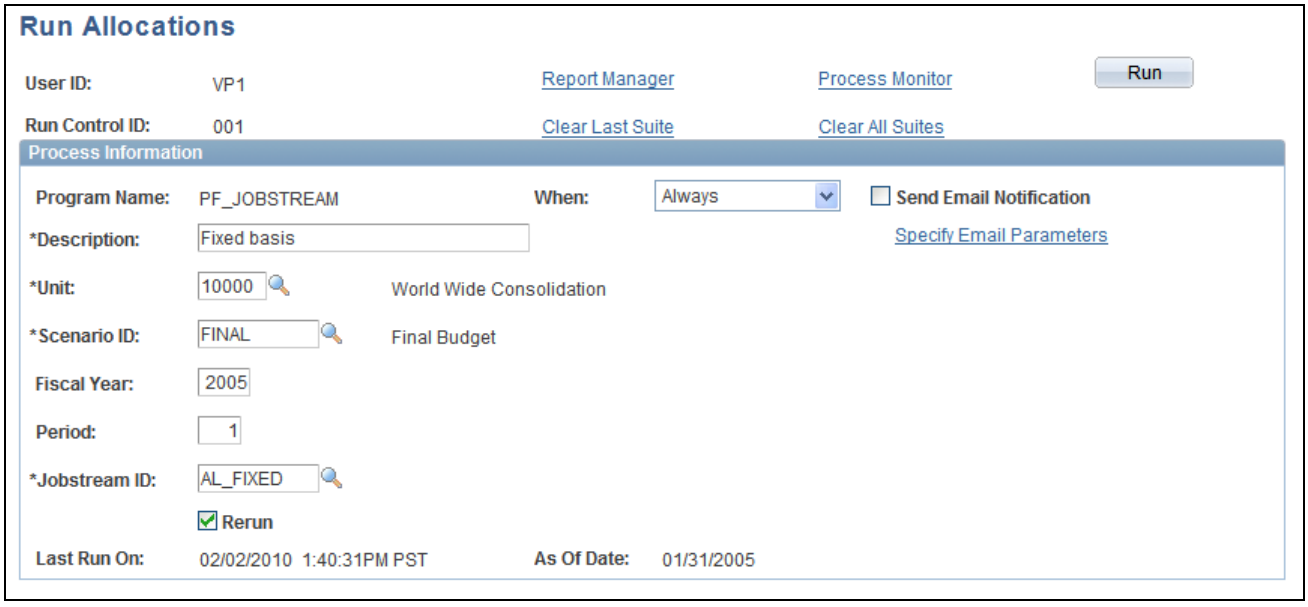
See Chapter 21, "Setting Up and Using Profit Manager," Processing and Posting Journals, page 516.

Sequence Number Enter a number, such as 100, for the first rule set in the job association. The next rule set to run would have a sequence of 200. The actual sequence number is not important; it represents the order in which you want to process rule sets. The sequence must be unique.

Rule Set Select the rule set or rule sets to include in the job. You create rule sets on the Data Manger Rule Set - Assignment Rule Sets page.

Processing Allocations

Access the Run Allocations page (EPM Foundation, Data Enrichment Tools, Allocation Manager, Run Allocations, Run Allocations)



Run Allocations page

As a last step to process your allocations, run the Allocation Manager engine.

As you run the jobstream:

- Use the Process Monitor to verify the status of your job.
- Review any error messages using the Error Messages component.

See Also

Chapter 20, "Streamlining Processing with Jobstreams," Viewing Engine Messages, page 494

Querying Allocations

Allocation Manager enables you to output allocation results to target batches or journals, or merge data directly to a ledger target table. Additionally, Allocation Manager sometimes uses PeopleSoft Enterprise Global Consolidations or Activity Based Management outputs as sources. Given these complex processes, it can be difficult to determine if the allocation output was calculated correctly based solely on viewing the results. As such, PeopleSoft EPM provides the new Allocation Manager Inquiry tool that enables you to review allocation audit data, from the target to source—specifically, you can review allocation targets, drill down to basis and source elements of an allocation, and view calculation rules.

The Allocation Manager Inquiry tool can also integrate with Global Consolidation and Activity Based Management allocation rules.

See Chapter 22, "Using Data Enrichment Tools," Drill to Source Page for an Activity Based Management Allocation, page 612.

See *PeopleSoft Enterprise Global Consolidations 9.1 PeopleBook*, "Analyzing Consolidations," Auditing Consolidation Data and *PeopleSoft Enterprise Activity-Based Management 9.1 PeopleBook*, "Reconciling Your Model and Analyzing Engine Output," Using Object Navigator and Model Analyzer.

Allocation Manager Audit Tables

The Allocation Manager Inquiry tool pulls data from the following EPM audit tables:

- Audit Target table (PF_AL_AUD_TGT)
- Audit Target Map table (PF_AL_AUD_TGMAP)
- Audit Basis table (PF_AL_AUD_BAS)
- Audit Basis Map table (PF_AL_AUD_BSMAP)
- Audit Source table (PF_AL_AUD_SRC)
- Audit Source Map table (PF_AL_AUD_SRMAP)
- Audit Common Dimension table (PF_AL_AUD_CDIM)
- Audit Map table (PF_AL_AUD_MAP)
- Audit Run log table (PF_AL_AUD_RUNLG)

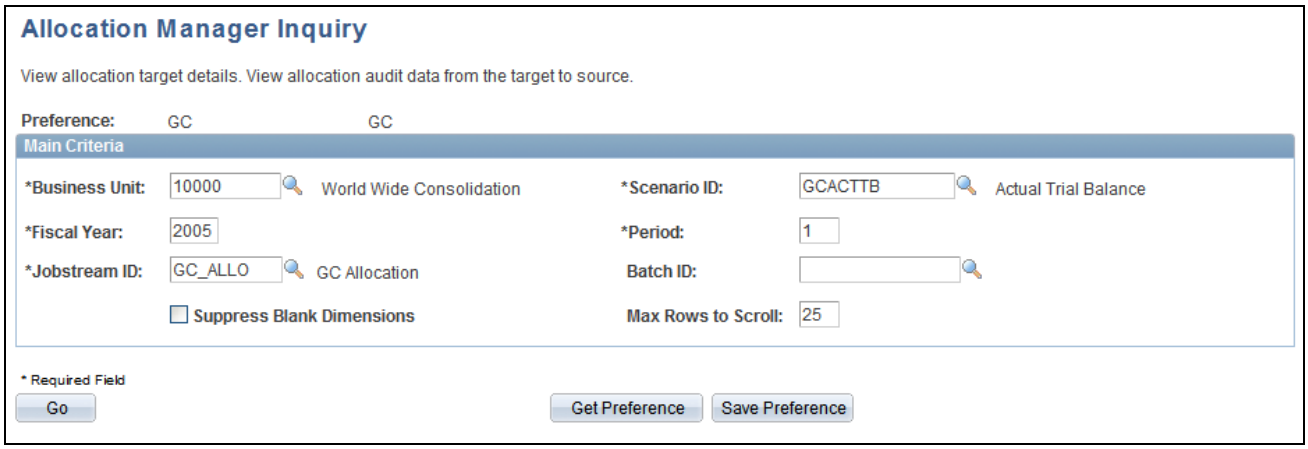
Pages Used to Query Target Allocations

Page Name	*Definition Name*	*Navigation*	*Usage*
Allocation Manager Inquiry - Audit Parameters page	PF_AL_INQUIRY1	EPM Foundation, Data Enrichment Tools, Allocation Manager, Allocation Manager Inquiry	Specify audit criteria for a specific allocation rule.

Page Name	Definition Name	Navigation	Usage
Allocation Manager Inquiry Results page	PF_AL_INQUIRY2	Click Go on the Allocation Manager Inquiry - Audit Parameters page	View allocation rule data that matched your criteria. This data is the starting point for the audit trail.
Allocation Manager Inquiry Results - Drill to Basis page	PF_AL_INQUIRY3	Click the Drill to Basis link on the Drill Details tab of the Allocation Manager Inquiry Results page	View basis rule details for an allocation.
Allocation Manager Inquiry Results - Drill to Source page	PF_AL_INQUIRY4	Click the Drill to Source link on the Drill Details tab of the Allocation Manager Inquiry Results page	View source rule details for an allocation.

Specifying the Query Criteria

Access the Allocation Manager Inquiry - Audit Parameters page (EPM Foundation, Data Enrichment Tools, Allocation Manager, Allocation Manager Inquiry).



Allocation Manager Inquiry - Audit Parameters page

Use the Allocation Manager Inquiry - Audit Parameters page to specify audit criteria for a specific allocation rule.

Select the business unit, scenario ID, and jobstream ID associated with allocation you want to audit. You must also enter the fiscal year and period for the allocation. You may select a batch ID associated with the allocation to audit, but this field is optional since you may want to inquire on multiple batches at one time or the target merged directly to the datamap and did not create a batch.

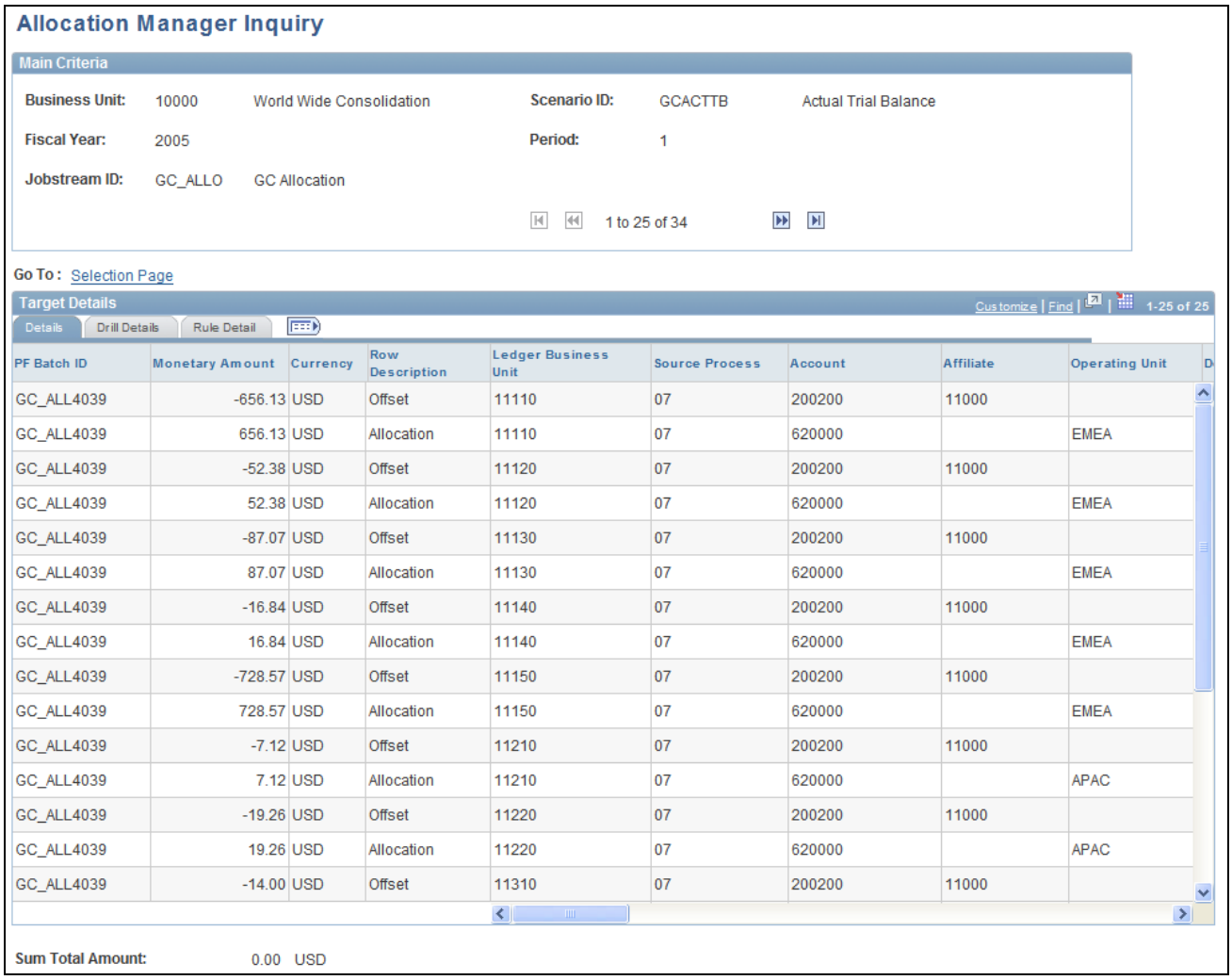
Suppress Blank Dimensions Select this check box to prevent dimension columns without values from displaying in the results grid.

Max Rows to Scroll Enter the maximum number of rows you want displayed in the results grid.

Go	Click to process the query parameters you provided and view the results grid.
Save Preference	Click to access the Save Inquiry Preference page and save the query parameters you input for future use. You can input a name and description for your preference. With the Default Preference check box you can set your parameters as the default values for the page, meaning the parameters automatically populate the parameters page when a user accesses it.
Get Preference	Click to access the Select Inquiry Preference page where you can select a preference that will automatically populate the parameters page with its values. With the Default Preference check box you can set the preference as the default for the page, meaning the parameters automatically populate the parameters page when a user accesses it.

Viewing Allocation Audit Data

Access the Allocation Manager Inquiry Results page (click Go on the Allocation Manager Inquiry - Audit Parameters page).



Allocation Manager Inquiry Results page

This page displays the target details associated with the allocation you selected for audit. Some of the columns displayed on this page are Monetary Amount, Row Description, and Ledger Business Unit. Values for the row description field include *Allocation*, *Offset*, and *Residual*.

If you did not specify a specific batch on the Allocation Manager Inquiry - Audit Parameters page, this page also displays the PF Batch ID.

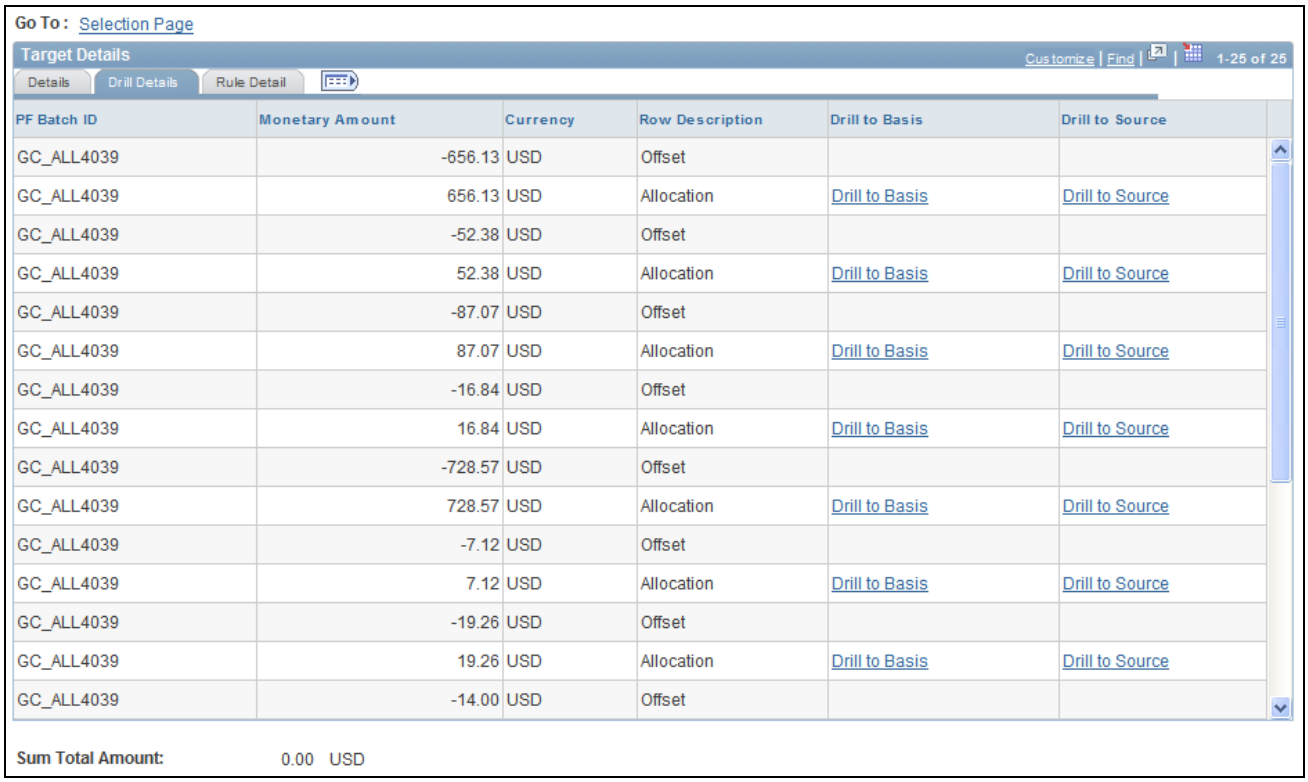
If you did not select the Suppress Blank Dimensions check box on the Allocation Manager Inquiry - Audit Parameters page, this page will display columns without values.

Click the Selection Page link to return to the Allocation Manager Inquiry - Audit Parameters page.

Note. The Audit Target (PF_AL_AUD_TGT) table is the source for all the column values on this page.

Drill Details Tab

Click the Drill Details tab to drill down to the basis or source information for an allocation.



Go To : Selection Page

Target Details Customize | Find | 1-25 of 25

Details | Drill Details | Rule Detail

PF Batch ID	Monetary Amount	Currency	Row Description	Drill to Basis	Drill to Source
GC_ALL4039	-656.13	USD	Offset		
GC_ALL4039	656.13	USD	Allocation	Drill to Basis	Drill to Source
GC_ALL4039	-52.38	USD	Offset		
GC_ALL4039	52.38	USD	Allocation	Drill to Basis	Drill to Source
GC_ALL4039	-87.07	USD	Offset		
GC_ALL4039	87.07	USD	Allocation	Drill to Basis	Drill to Source
GC_ALL4039	-16.84	USD	Offset		
GC_ALL4039	16.84	USD	Allocation	Drill to Basis	Drill to Source
GC_ALL4039	-728.57	USD	Offset		
GC_ALL4039	728.57	USD	Allocation	Drill to Basis	Drill to Source
GC_ALL4039	-7.12	USD	Offset		
GC_ALL4039	7.12	USD	Allocation	Drill to Basis	Drill to Source
GC_ALL4039	-19.26	USD	Offset		
GC_ALL4039	19.26	USD	Allocation	Drill to Basis	Drill to Source
GC_ALL4039	-14.00	USD	Offset		

Sum Total Amount: 0.00 USD

Drill Details tab

Click the Drill to Basis link to access the Allocation Manager Inquiry Results - Drill to Basis page and view basis details associated with the allocation.

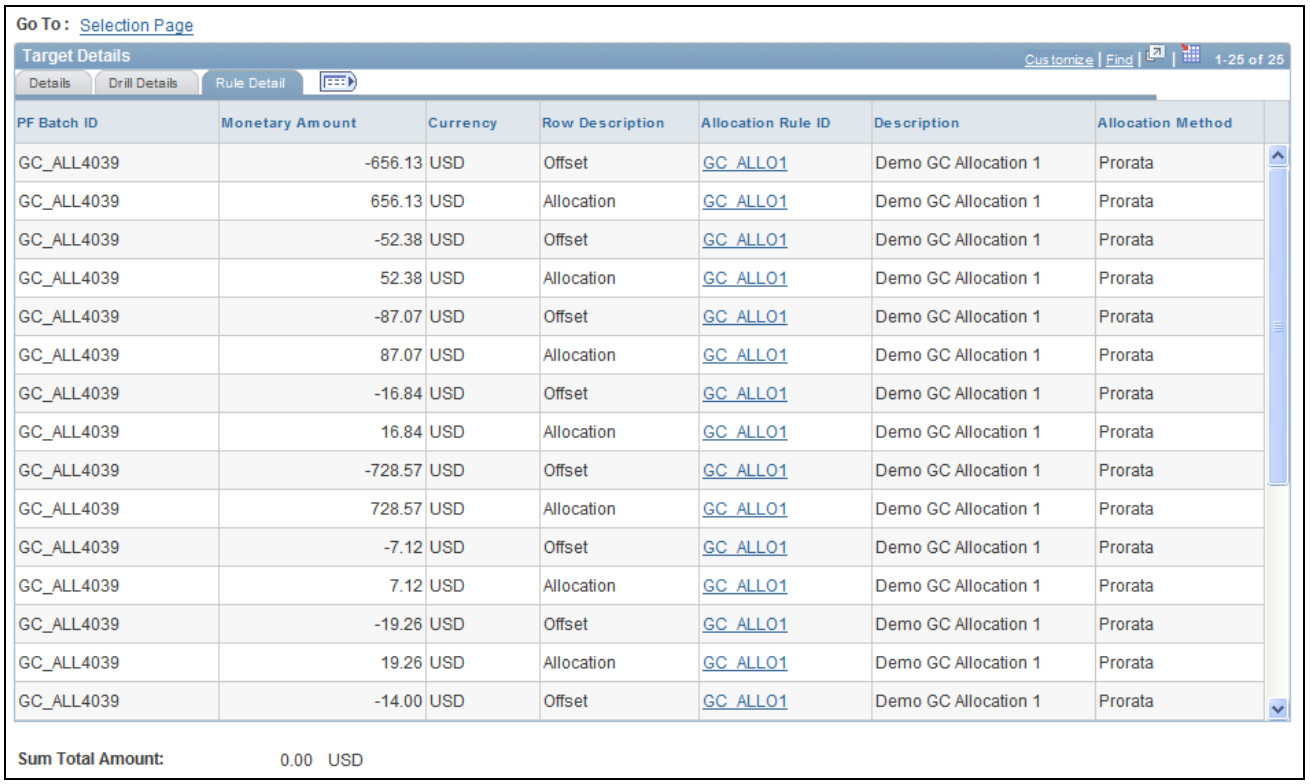
Click the Drill to Source link to access the Allocation Manager Inquiry Results - Drill to Source page and view source details associated with the allocation.

Click the Selection Page link to return to the Allocation Manager Inquiry - Audit Parameters page.

Note. Only rows labeled with *Allocation* for the Row Description column have the option to drill to the source and basis.

Rule Detail Tab

Click the Rule Detail tab to drill down to the rule information for an allocation.



Go To : Selection Page

Target Details — Customize | Find | 1-25 of 25

Details | Drill Details | Rule Detail

PF Batch ID	Monetary Amount	Currency	Row Description	Allocation Rule ID	Description	Allocation Method
GC_ALL4039	-656.13	USD	Offset	GC_ALLO1	Demo GC Allocation 1	Prorata
GC_ALL4039	656.13	USD	Allocation	GC_ALLO1	Demo GC Allocation 1	Prorata
GC_ALL4039	-52.38	USD	Offset	GC_ALLO1	Demo GC Allocation 1	Prorata
GC_ALL4039	52.38	USD	Allocation	GC_ALLO1	Demo GC Allocation 1	Prorata
GC_ALL4039	-87.07	USD	Offset	GC_ALLO1	Demo GC Allocation 1	Prorata
GC_ALL4039	87.07	USD	Allocation	GC_ALLO1	Demo GC Allocation 1	Prorata
GC_ALL4039	-16.84	USD	Offset	GC_ALLO1	Demo GC Allocation 1	Prorata
GC_ALL4039	16.84	USD	Allocation	GC_ALLO1	Demo GC Allocation 1	Prorata
GC_ALL4039	-728.57	USD	Offset	GC_ALLO1	Demo GC Allocation 1	Prorata
GC_ALL4039	728.57	USD	Allocation	GC_ALLO1	Demo GC Allocation 1	Prorata
GC_ALL4039	-7.12	USD	Offset	GC_ALLO1	Demo GC Allocation 1	Prorata
GC_ALL4039	7.12	USD	Allocation	GC_ALLO1	Demo GC Allocation 1	Prorata
GC_ALL4039	-19.26	USD	Offset	GC_ALLO1	Demo GC Allocation 1	Prorata
GC_ALL4039	19.26	USD	Allocation	GC_ALLO1	Demo GC Allocation 1	Prorata
GC_ALL4039	-14.00	USD	Offset	GC_ALLO1	Demo GC Allocation 1	Prorata

Sum Total Amount: 0.00 USD

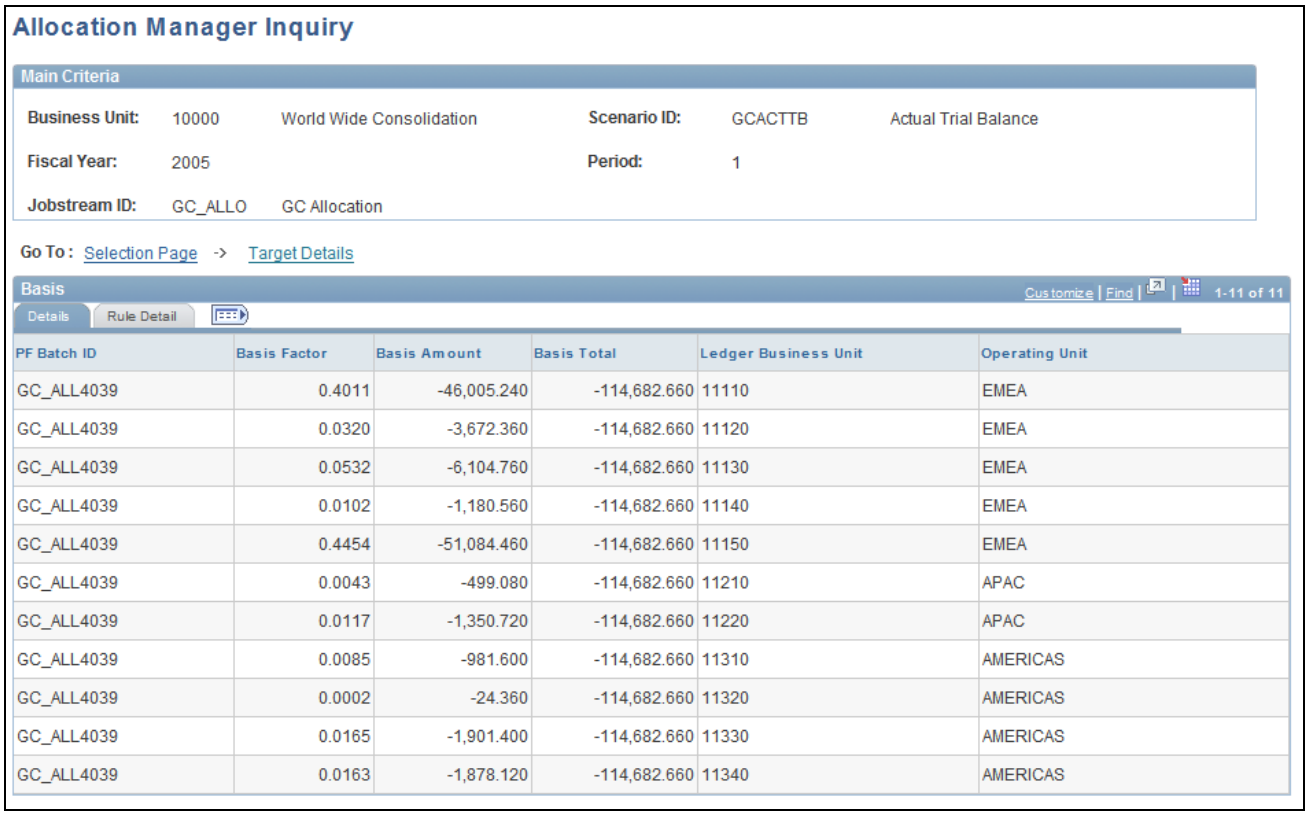
Rule Detail tab on the Allocation Manager Inquiry Results page

Click the allocation rule link in the Allocation Rule ID column to access the Method page and view allocation method details, such as allocation method type and period based allocations.

Click the Selection Page link to return to the Allocation Manager Inquiry - Audit Parameters page.

Drilling to the Allocation Basis

Access the Allocation Manager Inquiry Results - Drill to Basis page (click the Drill to Basis link on the Drill Details tab of the Allocation Manager Inquiry Results page).



Allocation Manager Inquiry

Main Criteria

Business Unit:	10000	World Wide Consolidation	Scenario ID:	GCACTTB	Actual Trial Balance
Fiscal Year:	2005		Period:	1	
Jobstream ID:	GC_ALLO	GC Allocation			

Go To : Selection Page -> Target Details

Basis — Customize | Find | 1-11 of 11

Details | Rule Detail

PF Batch ID	Basis Factor	Basis Amount	Basis Total	Ledger Business Unit	Operating Unit
GC_ALL4039	0.4011	-46,005.240	-114,682.660	11110	EMEA
GC_ALL4039	0.0320	-3,672.360	-114,682.660	11120	EMEA
GC_ALL4039	0.0532	-6,104.760	-114,682.660	11130	EMEA
GC_ALL4039	0.0102	-1,180.560	-114,682.660	11140	EMEA
GC_ALL4039	0.4454	-51,084.460	-114,682.660	11150	EMEA
GC_ALL4039	0.0043	-499.080	-114,682.660	11210	APAC
GC_ALL4039	0.0117	-1,350.720	-114,682.660	11220	APAC
GC_ALL4039	0.0085	-981.600	-114,682.660	11310	AMERICAS
GC_ALL4039	0.0002	-24.360	-114,682.660	11320	AMERICAS
GC_ALL4039	0.0165	-1,901.400	-114,682.660	11330	AMERICAS
GC_ALL4039	0.0163	-1,878.120	-114,682.660	11340	AMERICAS

Allocation Manager Inquiry Results - Drill to Basis page

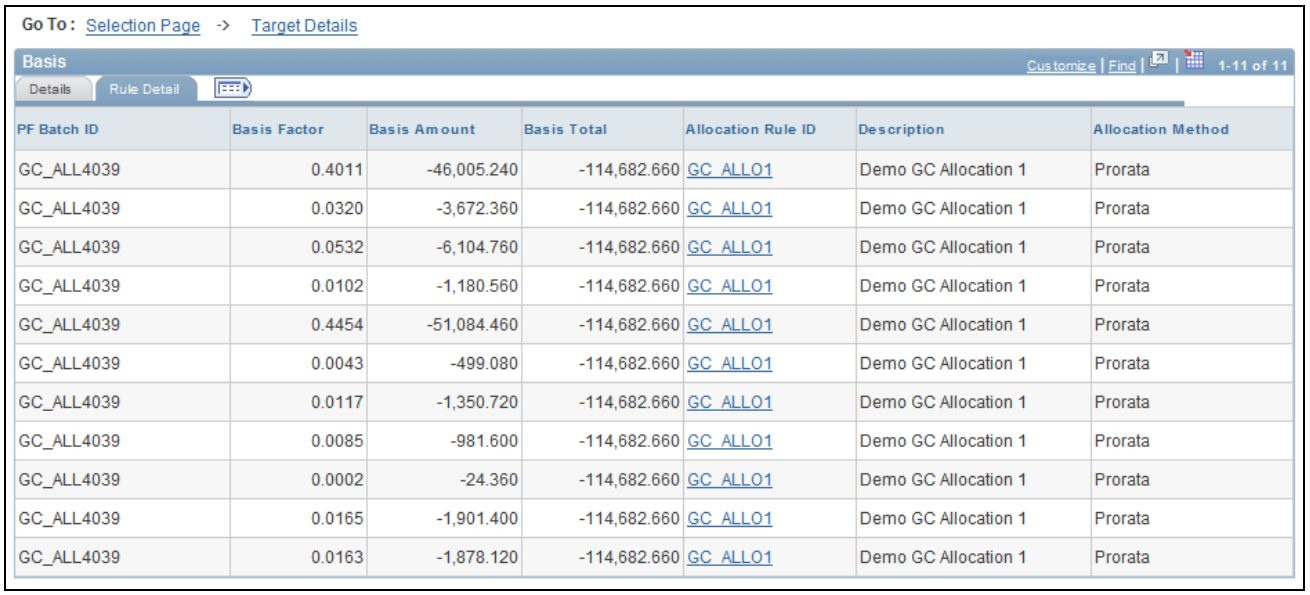
This page displays the basis details associated with the allocation you selected for audit. Some of the columns displayed on this page are Basis Factor, Basis Amount, and Basis Total. The basis total is the sum of the basis amounts taken from the Audit Basis (PF_AL_AUD_BAS) table. The basis factor is calculated as the ratio between the basis amount and basis total. If an adjustment was applied to the basis (and basis adjustment is not equal 1) an additional column called Adjustment Applied will appear in the grid and display the adjustment factor that was applied.

Click the Selection Page link to return to the Allocation Manager Inquiry - Audit Parameters page.

Click the Target Details link to return to the Allocation Manager Inquiry Results page.

Rule Detail Tab

Click the Rule Detail tab to view the basis rule summary for an allocation.



Go To : Selection Page -> Target Details

Basis — Customize | Find | 1-11 of 11

Details | Rule Detail

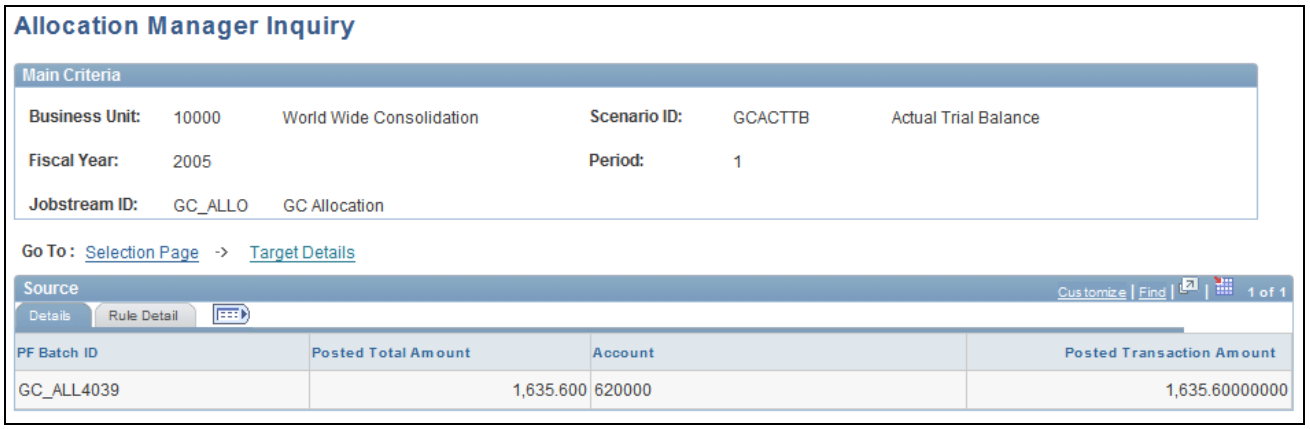
PF Batch ID	Basis Factor	Basis Amount	Basis Total	Allocation Rule ID	Description	Allocation Method
GC_ALL4039	0.4011	-46,005.240	-114,682.660	GC_ALLO1	Demo GC Allocation 1	Prorata
GC_ALL4039	0.0320	-3,672.360	-114,682.660	GC_ALLO1	Demo GC Allocation 1	Prorata
GC_ALL4039	0.0532	-6,104.760	-114,682.660	GC_ALLO1	Demo GC Allocation 1	Prorata
GC_ALL4039	0.0102	-1,180.560	-114,682.660	GC_ALLO1	Demo GC Allocation 1	Prorata
GC_ALL4039	0.4454	-51,084.460	-114,682.660	GC_ALLO1	Demo GC Allocation 1	Prorata
GC_ALL4039	0.0043	-499.080	-114,682.660	GC_ALLO1	Demo GC Allocation 1	Prorata
GC_ALL4039	0.0117	-1,350.720	-114,682.660	GC_ALLO1	Demo GC Allocation 1	Prorata
GC_ALL4039	0.0085	-981.600	-114,682.660	GC_ALLO1	Demo GC Allocation 1	Prorata
GC_ALL4039	0.0002	-24.360	-114,682.660	GC_ALLO1	Demo GC Allocation 1	Prorata
GC_ALL4039	0.0165	-1,901.400	-114,682.660	GC_ALLO1	Demo GC Allocation 1	Prorata
GC_ALL4039	0.0163	-1,878.120	-114,682.660	GC_ALLO1	Demo GC Allocation 1	Prorata

Rule Detail tab on the Drill to Basis page

Click the allocation rule link displayed in the Allocation Rule ID column to access the Basis page and view basis rule details, such as adjustments, constraints. or if it is a fixed basis.

Drilling to the Allocation Source

Access the Allocation Manager Inquiry Results - Drill to Source page (click the Drill to Source link on the Drill Details tab of the Allocation Manager Inquiry Results page).



Allocation Manager Inquiry

Main Criteria

Business Unit: 10000 World Wide Consolidation Scenario ID: GCACTTB Actual Trial Balance

Fiscal Year: 2005 Period: 1

Jobstream ID: GC_ALLO GC Allocation

Go To : Selection Page -> Target Details

Source — Customize | Find | 1 of 1

Details | Rule Detail

PF Batch ID	Posted Total Amount	Account	Posted Transaction Amount
GC_ALL4039	1,635.600	620000	1,635.60000000

Allocation Manager Inquiry Results - Drill to Source page

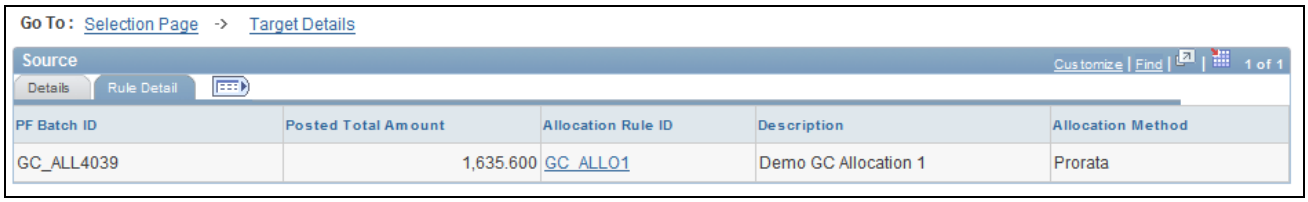
This page displays the source details associated with the allocation you selected for audit. Some of the columns displayed on this page are Posted Total Amount and Posted Transaction Amount. If an adjustment was applied to the source (and source adjustment is not equal 1) an additional column called Adjustment Applied will appear in the grid and display the adjustment factor that was applied.

Click the Selection Page link to return to the Allocation Manager Inquiry - Audit Parameters page.

Click the Target Details link to return to the Allocation Manager Inquiry Results page.

Rule Detail Tab

Click the Rule Detail tab to view the source rule summary for an allocation.



Go To : Selection Page -> Target Details

Source — Customize | Find | 1 of 1

Details | Rule Detail

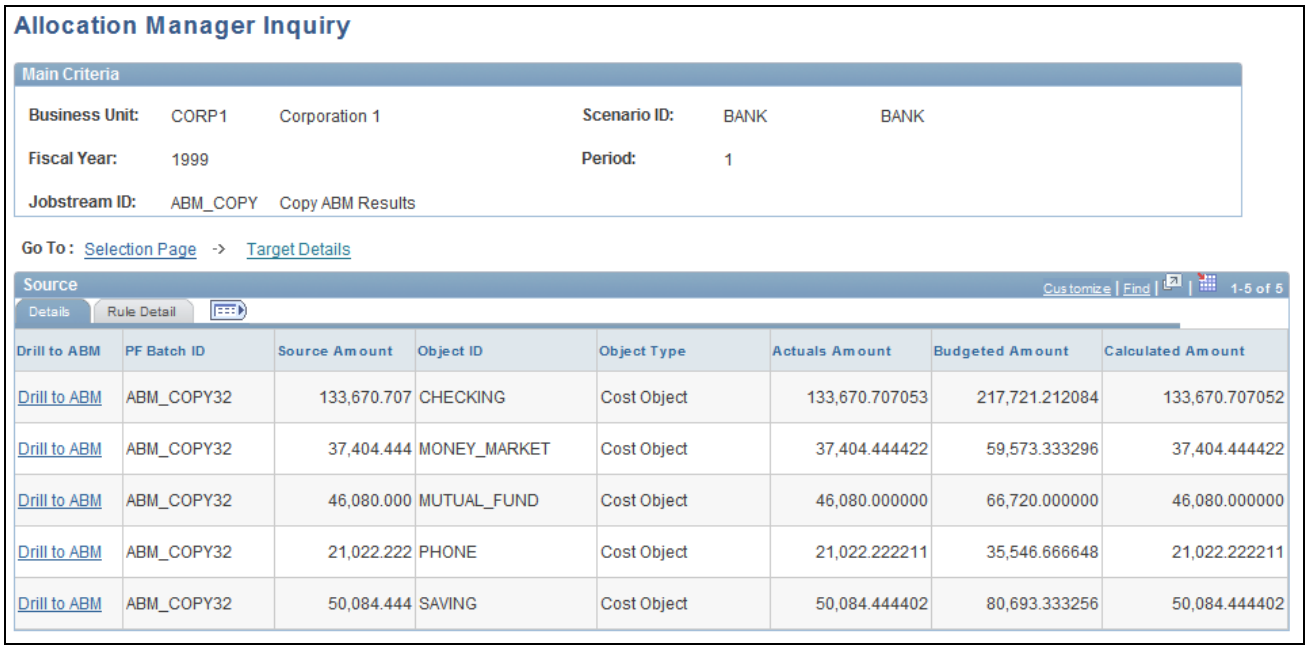
PF Batch ID	Posted Total Amount	Allocation Rule ID	Description	Allocation Method
GC_ALL4039	1,635.600	GC_ALLO1	Demo GC Allocation 1	Prorata

Rule Detail tab on the Drill to Source page

Click the link displayed in the Allocation Rule ID column to access the Source page and view source rule details, such as adjustments, constraints, or if it is a fixed source.

Drill to Source Page for an Activity Based Management Allocation

The Allocation Manager Inquiry Results - Drill to Source page changes slightly in appearance and functionality when you are working with an ABM allocation.



Allocation Manager Inquiry

Main Criteria

Business Unit:	CORP1	Corporation 1	Scenario ID:	BANK	BANK
Fiscal Year:	1999		Period:	1	
Jobstream ID:	ABM_COPY	Copy ABM Results			

Go To : Selection Page -> Target Details

Source — Customize | Find | 1-5 of 5

Details | Rule Detail

Drill to ABM	PF Batch ID	Source Amount	Object ID	Object Type	Actuals Amount	Budgeted Amount	Calculated Amount
Drill to ABM	ABM_COPY32	133,670.707	CHECKING	Cost Object	133,670.707053	217,721.212084	133,670.707052
Drill to ABM	ABM_COPY32	37,404.444	MONEY_MARKET	Cost Object	37,404.444422	59,573.333296	37,404.444422
Drill to ABM	ABM_COPY32	46,080.000	MUTUAL_FUND	Cost Object	46,080.000000	66,720.000000	46,080.000000
Drill to ABM	ABM_COPY32	21,022.222	PHONE	Cost Object	21,022.222211	35,546.666648	21,022.222211
Drill to ABM	ABM_COPY32	50,084.444	SAVING	Cost Object	50,084.444402	80,693.333256	50,084.444402

Allocation Manager Inquiry Results - Drill to Source page for an ABM allocation

This page displays the source details associated with the ABM allocation you selected for audit. Some of the ABM-specific columns displayed on this page are Object ID, Object Type and Budgeted Amount.

Clicking the link displayed in the Drill to ABM column transfers you to the Object Navigator Inquiry tool found in ABM. Using the ABM tool you can further drill down to the model with its resources, activities, and cost objects, as well as view their sources and targets.

See *PeopleSoft Enterprise Activity-Based Management 9.1 PeopleBook*, "Reconciling Your Model and Analyzing Engine Output," Using Object Navigator and Model Analyzer.

Chapter 23

Creating XBRL Instance Documents in EPM

This document provides an overview of XBRL reporting using PeopleSoft EPM, and discusses how to:

- Set up XBRL definitions in EPM.
- Upload and maintain XBRL facts and taxonomy elements.
- Create XBRL instance documents.

Understanding XBRL Reporting Using PeopleSoft EPM

This chapter discusses XBRL reporting functionality, which enables you to meet the XBRL financial reporting requirements and create XBRL Instance documents per the XBRL Specifications 2.1, and includes:

- • Pages that enable you to define the components for an XBRL Instance document.
- • Pages that enable you to set up and define taxonomy elements and the required sections of an XBRL Instance document.
- • An import utility to upload taxonomy elements and financial report facts.
- • An application engine process that generates the XBRL Instance report.

This document discusses how to set up the new XBRL Reporting functionality and create XBRL instance documents for submission to a specified reporting entity. Examples used are for US GAAP.

Note. The functionality deals only with XBRL encoding. This means that you may have generated your financial reports in any source system, (PeopleSoft Enterprise General Ledger or PeopleSoft Enterprise Global Consolidations, for example) or using any reporting tool (PS nVision, for example) or any other third party tool.

XBRL Reporting Requirements and Reference Resources

In May 2008, the United States Security Exchange Commission (SEC) issued rules to adopt eXtensible Business Reporting Language (XBRL), the financial reporting version of XML, as the standard to meet financial reporting requirements.

The SEC's proposed schedule would require companies using U.S. Generally Accepted Accounting Principles (US GAAP) with a worldwide public float over $5 billion to make financial disclosures using interactive data formatted in eXtensible Business Reporting Language (XBRL) for fiscal periods ending in late 2008. If adopted, the first interactive data provided under the new rules would be made public in early 2009. The remaining companies using U.S. GAAP would provide this disclosure over the following two years. Companies using International Financial Reporting Standards (IFRS) would provide this disclosure for fiscal periods ending in late 2010. The disclosure would be provided as additional exhibits to annual and quarterly reports and registration statements. Companies also would be required to post this information on their websites.

The required tagged disclosures would include companies' primary financial statements, notes, and financial statement schedules. Initially, companies would tag notes and schedules as blocks of text, and later, they would provide tags for the details within the notes and schedules.

The following table lists references and resources for more information about the SEC rulings and XBRL requirements.

Documents	***Location***
XBRL Specifications 2.1	http://www.xbrl.org/Specification/XBRL-RECOMMENDATION-2003-12-31+Corrected-Errata-2008-07-02.htm/
Financial Reporting Instance Standards 1.0	http://www.xbrl.org/technical/guidance/FRIS-PWD-2004-11-14.htm
Tutorial and in-depth technical training on XBRL	http://www.xbrl.org/EducationAndTraining/
Education and Training	http://www.xbrl.org/EducationAndTraining/
XBRL US GAAP Taxonomy 1.0	http://www.xbrl.org/FRTaxonomies/
XBRL US GAAP Taxonomy Preparers Guide	http://xbrl.us/Documents/PreparersGuide.pdf

Note. This documentation assumes that you are familiar with the XBRL instance document requirements; it does not describe the XBRL specifications, reporting standards, or taxonomy structure. Please refer to the reference materials in the previous table for detailed information about XBRL.

Using the XBRL Reporting Features

The following list outlines the typical activities that you'll complete to generate XBRL financial statements:

1. Prepare Financial Statements and map elements to required taxonomy.

   These steps are completed outside of the PeopleSoft EPM application, using financial applications, spreadsheets, and the XBRL websites to prepare and tag the data. Typically this phase includes the following tasks:

   a. Close the accounting period.

   b. Generate, review, and finalize financial reports (which includes formatting & rounding of amounts).

   c. Map the line items of your source financial reports to GAAP Taxonomy elements, using a spreadsheet program or similar tool, and using applicable XBRL taxonomy websites.

      The number of elements that are applicable to your organization is likely to be a small subset of all taxonomy elements. For example, there are 12,000 elements defined for US GAAP. A US GAAP user will access the US XBRL website and choose their industry entry point to review and copy the taxonomy elements (and some of their attributes) applicable to their reporting.

      Note. The US XBRL website provides a taxonomy viewer to search and look up taxonomy elements for financial report line items and standard footnotes.

2. Set up XBRL Definitions in EPM. This information is discussed in the section Setting Up XBRL Definitions in EPM.

3. Upload XBRL taxonomy and financial element facts, and create footnotes.

   These steps are discussed in the section Uploading and Maintaining XBRL Facts and Taxonomy Elements.

4. Create an XBRL instance report.

   This step is discussed in the section Creating XBRL Instance Documents.

Setting Up XBRL Definitions in EPM

This section provides an overview of XBRL instance document components, defines pages used, and discusses how to:

- Define a namespace set
- Define taxonomy
- Define an Instance Header
- Define a reporting entity
- Define an instance report.
- Define a report set.
- Define XBRL units of measure.
- Define context.

Understanding XBRL Instance Document Components

XBRL instance documents contain facts reported by a specific entity, at a specific point in time, in a specific currency using GAAP guidelines like US GAAP, IFRS, and so on.

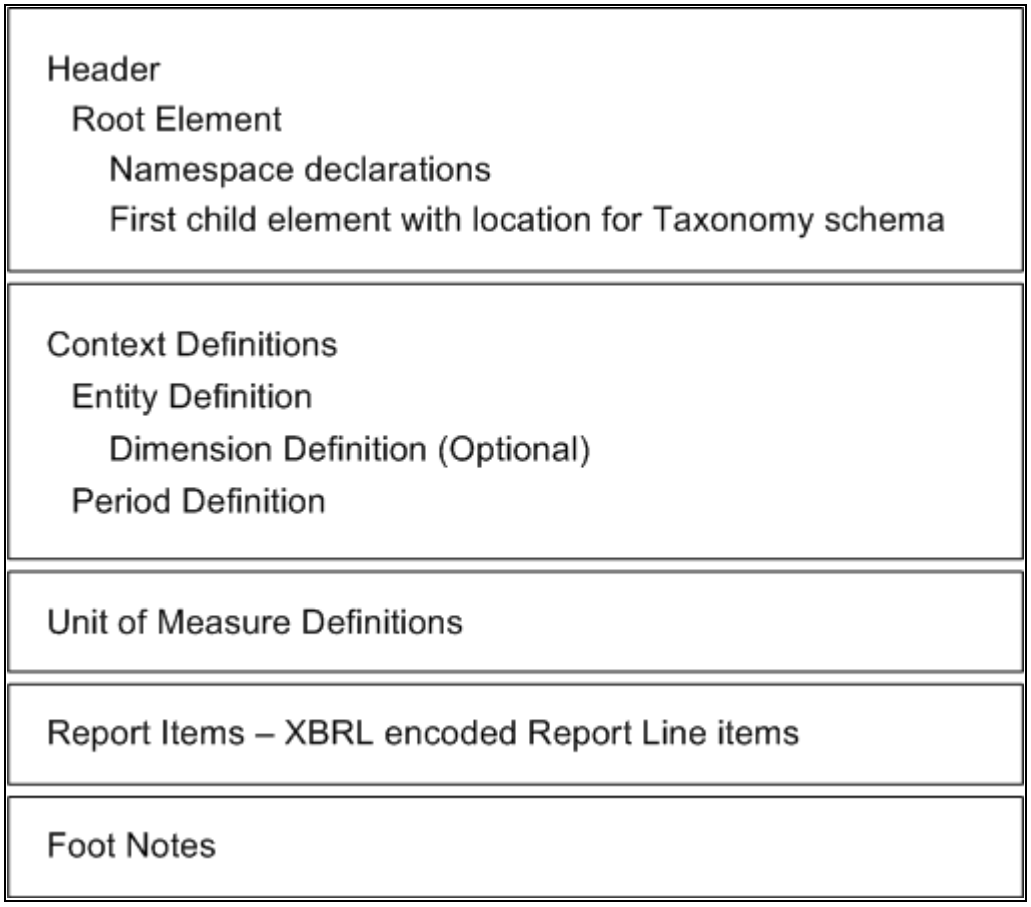
A GAAP taxonomy document contains definitions for the accounting concepts and their relationships for a specific GAAP like US GAAP. You can visualize it as GL account definitions and its reporting tree for US GAAP. Technically, it includes an xml schema definition and other supporting xml files.

There can be several instance documents for a taxonomy document. For example, you can have two balance sheet instances (reports) for a legal reporting entity – one for year 2009 and another for year 2010. Or you could have balance sheet instances for a parent corporation and all of its subsidiaries.

XBRL instance documents are XML documents and need to be well-formed (as defined by the XML schema) and valid (as defined by the taxonomy schema).

XBRL Shell and Sample Structure

The basic "shell" of an XBRL instance document includes the following structure:



XBRL Shell and Sample Structure

The following table provides an example of an XBRL instance. Table breaks separate each "shell" section.

XBRL Tag	*Description*
<?xml version="1.0"?>	All XML documents start with a prolog. This is an XML Identifier. This line is the required beginning line for every XML document, regardless of its type. Additional data can be inserted but this is the minimum required information. This line indicates that it is an XML document based on XML Recommendation 1.0. As a minimum, it must be included exactly as shown.
<xbrl	The root element is <xbrl> This is followed by namespace declarations for xml and for the taxonomy referenced in the instance document.
xmlns="http://www.xbrl.org/2003/instance"	This is the namespace for all XBRL instance documents. You can access the schema using this URL. (The schema defines the structure and format of their instances).
xmlns:link="http://www.xbrl.org/2003/linkbase"	This identifies the location of XBRL Link bases - Anything referring to the role of linkbases within XBRL is defined here and any reference to it will be prefixed with the alias, "link:"
xmlns:xlink="http://www.w3.org/1999/xlink"	This is the location of BASE xml link bases. It refers to the roles of link bases specified within the XLink specifications.
xmlns:us-gaap="http://xbrl.us/us-gaap/2009-01-31" >	This is the US GAAP Taxonomy Namespace Reference
<link:schemaRef xlink:type="simple" xlink:href=" http://taxonomies.xbrl.us/us-gaap/2009/elts/us-gaap-std-2009-01-31.xsd"/>	In an XBRL instance, the schemaRef element points to a taxonomy schema. Every XBRL instance must contain at least one schemaRef element. The schemaRef element must occur as a child element of an xbrl element and must occur before other children of the xbrl root element, in document order.

<Context id="Asof_June30" >	The context element contains information about the entity being described and the reporting period, all of which are necessary for understanding a business fact captured as an XBRL item. The context element has an attribute, ID = '"Asof_June30" in this example
<entity>	The entity element.

<identifier scheme="http://www.sec.gov/">1234567890</identifier>	The entity is identified using 2 attributes: • Scheme – the URL for SEC • ID – the 10 digit central index key (CIK) assigned by SEC (The entity may also be identified using a URL for the stock exchange and the ticker symbol of the company).
</entity>	Closing tag.
<period>	The period element.
<instant>2008-06-30</instant>	Period type is instant and is 'As of date', June 30, 2008.
</period>	Closing tag.
</Context>	Closing tag.

<unit> ID="usd"	Unit Element with a ID attribute.
<measure> ISO4217:USD </measure>	Measure is the sub-element. The unit of measure is US dollars and takes its meaning from ISO4217.
</unit>	Closing tag.

<us-gaap:AssetsCurrent decimal="INF" unitRef="usd" contextRef=" Asof_June30">727</us-gaap:AssetsCurrent >	The element details are comprised of the following: • us-gaap – the prefix(alias) refers to the US GAAP Taxonomy • decimal="INF" – The fact value is exact (not rounded) • AssetsCurrent – the element name The element has 2 reference attributes: • unitRef="usd" – refers to the unit Id defined above • contextRef=" Asof_June30" – refers to context Id defined above 727 – is the fact value for the element
< us-gaap:liabilities precision="3" unitRef="usd" contextRef=" Asof_June30">635</ci:liabilities>	The element details for US GAAP liabilities.

</xbrl>	End of instance document. Closing tag for the root element

You will use the pages described in the following section to establish the components for an XBRL instance report.

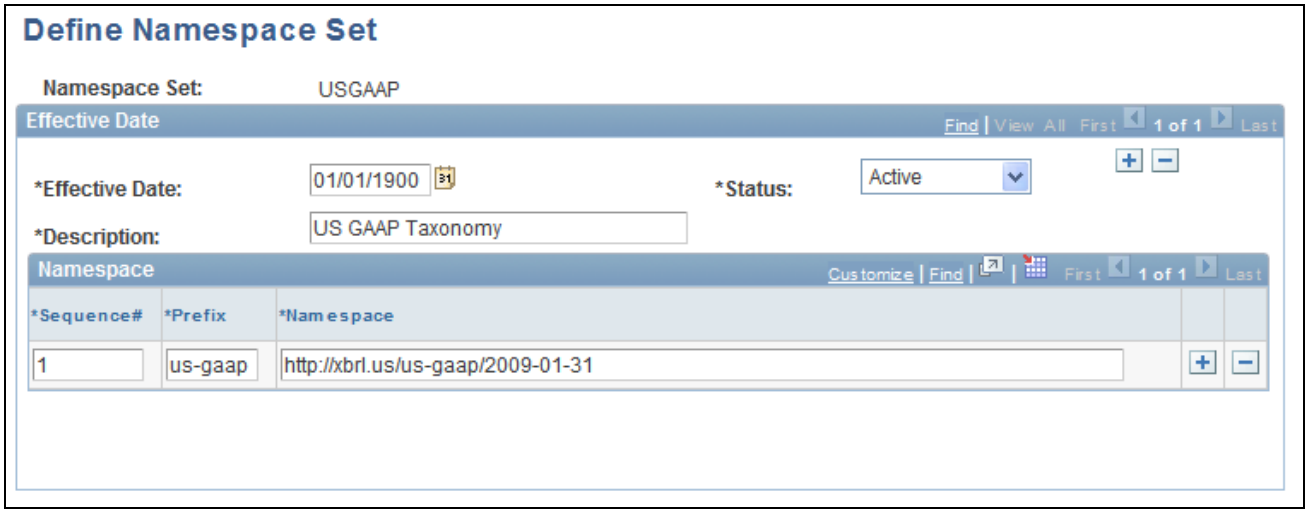
Pages Used to Set up Reporting Definitions

Page Name	*Definition Name*	*Navigation*	*Usage*
Define Namespace Set	XBRL_NS_SET	EPM Foundation, XBRL Reporting, Define XBRL Taxonomy, Define Namespace Set	Define XML namespaces. These will be declared in the root element of the Instance. Some namespaces are specific to XML and some to a specific taxonomy. The sets are re-usable and are referenced in the Instance template
Define Taxonomy	XBRL_TAXONOMY	EPM Foundation, XBRL Reporting, Define XBRL Taxonomy, Define Taxonomy	Define taxonomy attributes, including its name space and the physical location of its Schema. These attributes are used in the Instance document.
Define Instance Header	XBRL_INST_HDR	EPM Foundation, XBRL Reporting, Define XBRL Taxonomy, Define Instance Header	Define an XBRL instance document header.
Define Reporting Entity	XBRL_ENTITY	EPM Foundation, XBRL Reporting, Define Reporting, Define Reporting Entity	Define a reporting entity.
Define Instance Report	XBRL_RPT_DEFN	EPM Foundation, XBRL Reporting, Define Reporting, Define Instance Report	Define a report ID.
Define Report Set	XBRL_RPT_SET	EPM Foundation, XBRL Reporting, Define Reporting, Define Report Set	Create Report Sets, which are a grouping of report IDs.

Page Name	Definition Name	Navigation	Usage
Define XBRL Unit of Measure	XBRL_UOM	EPM Foundation, XBRL Reporting, Define Reporting, Define XBRL Units of Measure	Create Unit of Measure (UOM) definitions for XBRL reporting.
Define Context	XBRL_CONTEXT	EPM Foundation, XBRL Reporting, Define Reporting, Define Context	Define context IDs.

Defining a Namespace Set

Access the Define Namespace Set page (EPM Foundation, XBRL Reporting, Define XBRL Taxonomy, Define Namespace Set).



Define Namespace Set page

Use this page to define the namespaces to include in a namespace set. Namespaces are declared in the root element of the Instance document. Some namespaces are specific to XML and some to a specific taxonomy. You will reference a namespace set when you define an instance header, and when you define taxonomy, so typically, you will define at least two namespace sets.

Description Enter the description for the namespace set, for example, US GAAP Taxonomy.

Namespace Group Box

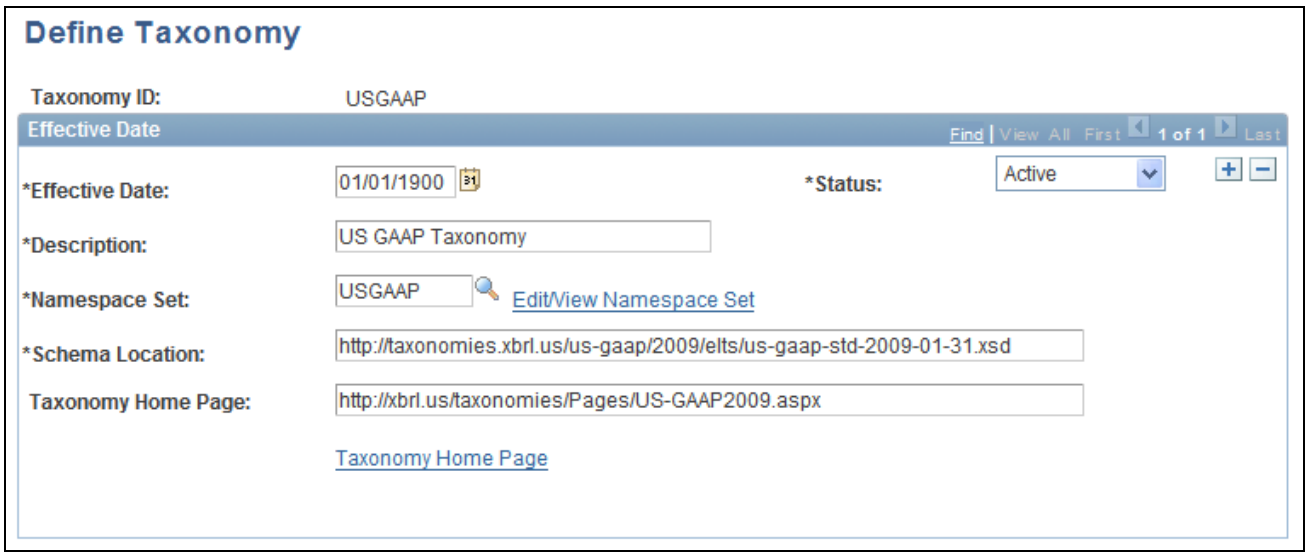
Complete the following fields for each namespace that you add to the Namespace grid.

Sequence Enter a value for each namespace within the namespace set. This must be unique; it functions as an identifier for each namespace.

Prefix	Enter a shorthand sequence of letters for a namespace; us-gaap, for example, is a recommended prefix for the namespace http://xbrl.us/us-gaap/2008-01-31.
Namespace	Enter the Universal Resource Identifier (URI) that identifies the namespace.
Add Button	Click the Add button to specify additional namespaces within this namespace set.

Defining the Taxonomy

Access the Define Taxonomy page (EPM Foundation, XBRL Reporting, Define XBRL Taxonomy, Define Taxonomy).



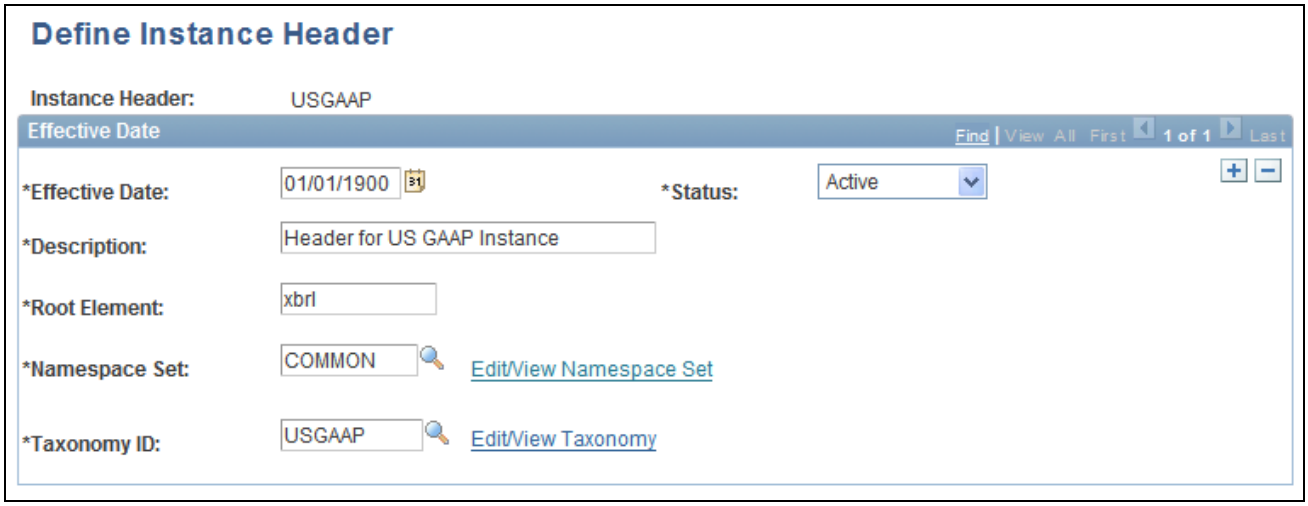
Define Taxonomy page

Use this page to define a taxonomy. You reference the Taxonomy ID when you define an instance header.

Description	Enter a description for the taxonomy.
Namespace Set	Select the namespace set that is associated with this taxonomy. Click the Edit/View Namespace Set link to access the Define Namespace Set page, where you can edit or view the namespace set.
Schema Location	Enter the Universal Resource Identifier (URI) that identifies the physical location of the taxonomy schema.
Taxonomy Home Page	Enter the URL for the taxonomy homepage.
Taxonomy Home Page link	Click to view the home page for the specified taxonomy.

Defining an Instance Header

Access the Define Instance Header page (EPM Foundation, XBRL Reporting, Define XBRL Taxonomy, Define Instance Header).



Define Instance Header page

This page captures the following attributes of an XBRL instance:

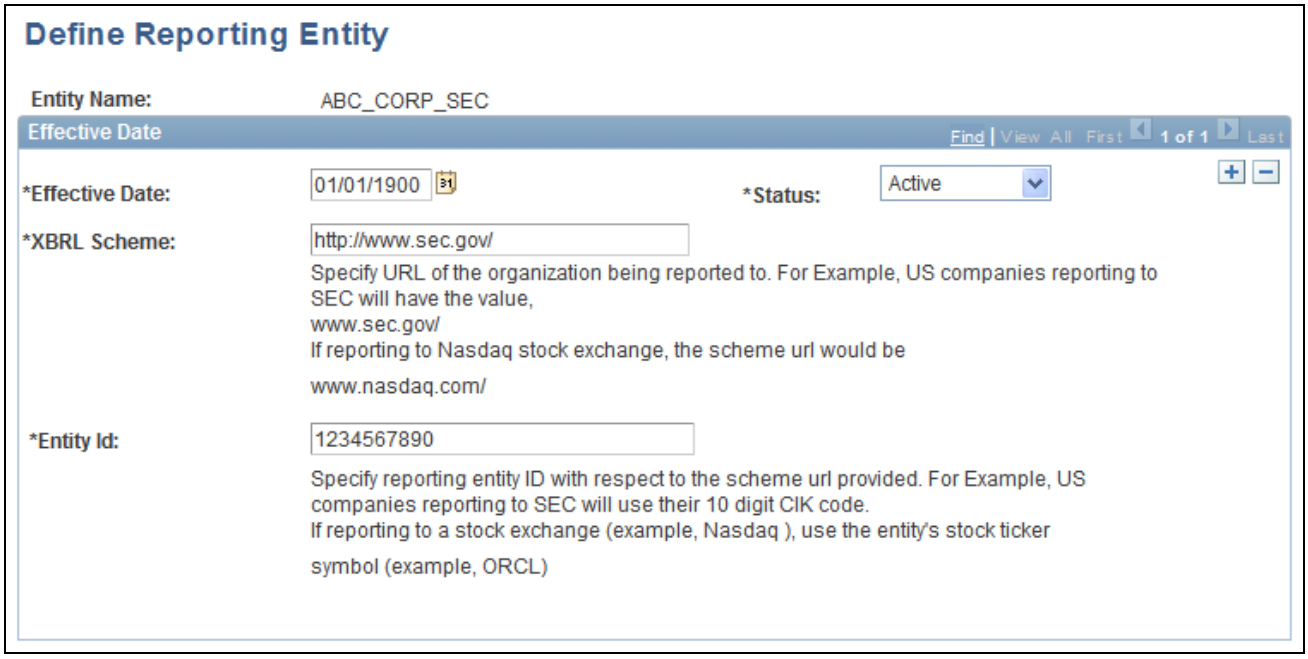
- The root element (currently "xbrl").
- The namespace set
- The taxonomy ID.

You reference an instance header when you create an XBRL instance.

Instance Header and Description	Enter a unique name for the XBRL instance header, and its description.
Root Element	Specify the root element for the instance document.
Namespace Set	Specify the namespace set to associate with the instance document. You establish namespace sets using the Define Namespace Set page.
Edit/View Namespace Set	Click to access the Define Namespace Set page for this namespace, where you can review the Namespace Set definition.
Taxonomy ID	Specify the taxonomy to associate with this instance header. You establish taxonomy IDs using the Define Taxonomy page.
Edit/View Taxonomy	Click to access the Define Taxonomy page for the specified taxonomy, where you can review or edit the taxonomy definition.

Defining a Reporting Entity

Access the Define Reporting Entity page (EPM Foundation, XBRL Reporting, Define Reporting, Define Reporting Entity).



Define Reporting Entity page

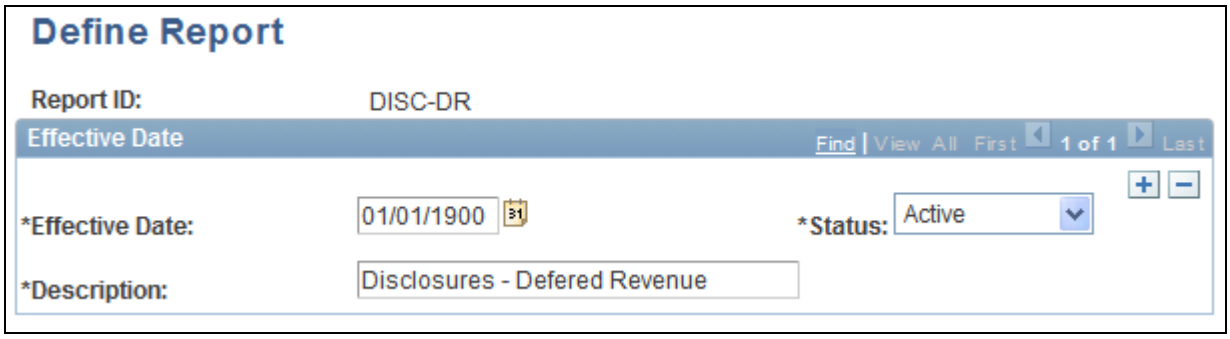
Entity Name Enter a unique name for the reporting entity.

XBRL Scheme Enter the URL of the organization you are reporting to.

Entity ID Enter your organization's Entity ID for the organization to which you are reporting.

Defining an Instance Report

Access the Define Instance Report page (EPM Foundation, XBRL Reporting, Define Reporting, Define Instance Report).



Define Instance Report page

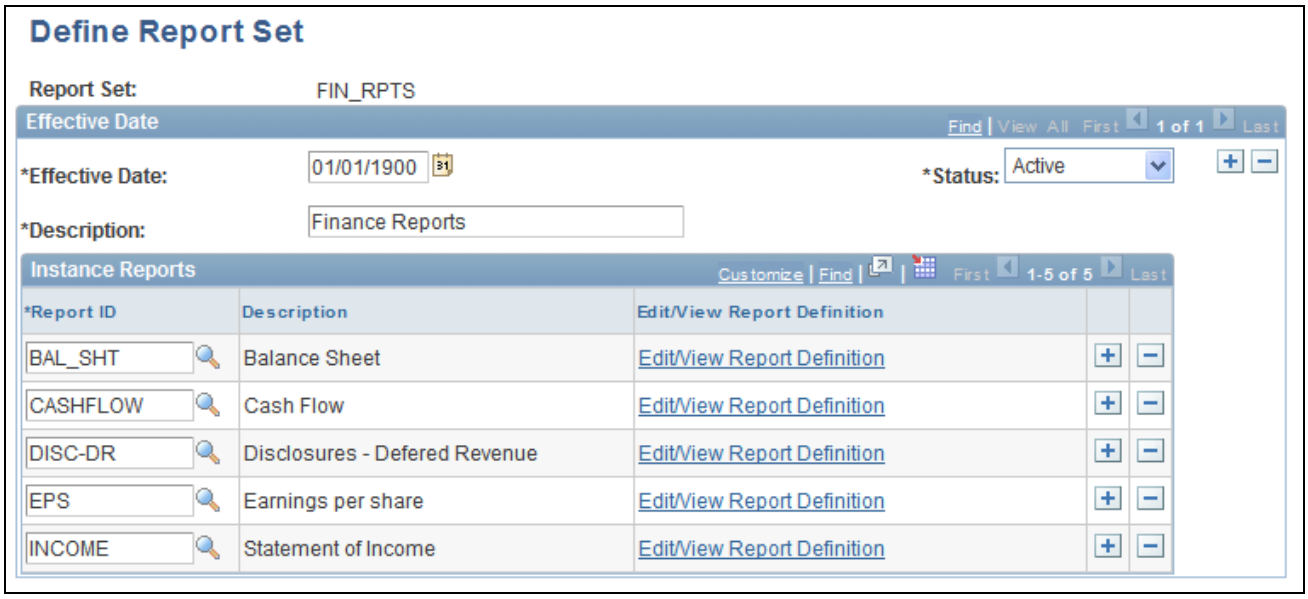
Report ID Enter a unique identifier for the instance report.

Description Enter the description to use for the instance report.

The system creates an instance document for either a single report ID or a report set.

Defining a Report Set

Access the Define Report Set page (EPM Foundation, XBRL Reporting, Define Reporting, Define Report Set).



Define Report Set page

Report Set Enter a unique name for the report set.

Description Enter a description for the report set.

Instance Reports Group Box

Specify the report IDs to include in this report set. You can add additional rows to include more than one report ID in the report set.

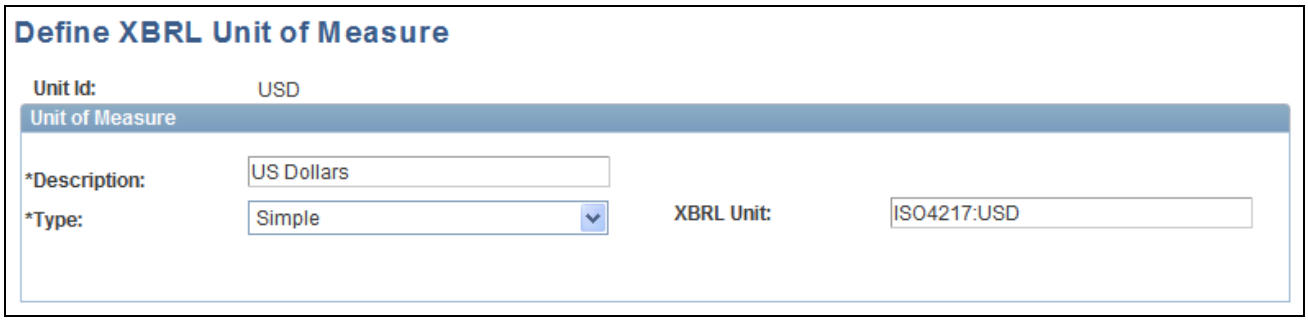
Report ID Specify the report ID to include. You establish report IDs using the Define Instance Report page.

Edit/View Report Definition Click this link to access the Define Instance Report page, where you can review or modify the report definition.

The system creates an instance document for either a single report ID or a report set.

Defining XBRL Unit of Measure

Access the Define XBRL Unit of Measure page (EPM Foundation, XBRL Reporting, Define Reporting, Define XBRL Units of Measure).



Define XBRL Unit of Measure page

Unit ID Enter a unique identifier for the XBRL unit of measure. The unit of measure IDs are referenced in the instance facts.

Description Enter a description for the unit of measure.

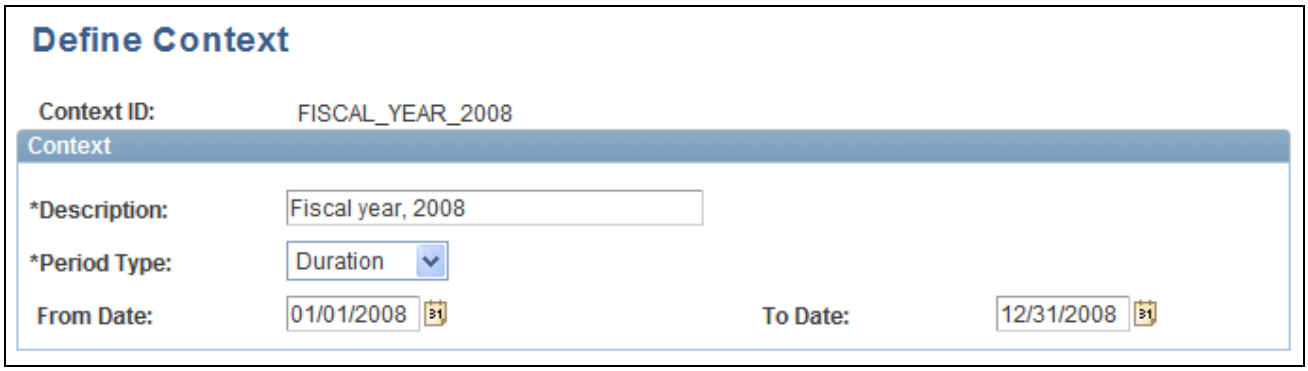
Type Specify the unit of measure type, either *Simple* or *Fraction.*

Currency is an example of a Simple type of unit of measure; earnings per share is an example of a Fraction type of unit of measure.

For a fraction type of unit of measure, enter the values for Numerator and Denominator. For a simple type of unit of measure, select *Simple* and enter the value for XBRL Unit.

Defining Context

Access the Define Context page (EPM Foundation, XBRL Reporting, Define Reporting, Define Context).



Define Context page

For each context you define, complete the following fields.

Context ID Enter the context identifier. PeopleSoft recommends using something that is easily interpreted, such as DefRev_Jun30_2008.

Description Enter a description for the context.

Period Type Specify the timeframe for the context.

Select *Instant* to define a specific point in time, and select the date in the As of Date field. This option would apply to a balance sheet report, for example.

Select *Duration* to define a date range, and select the starting and ending dates in the From Date and To Date fields. This option would apply to income and cash flow reports, for example.

The context ID within an instance is for a unique combination of entity, segment and time dimensions. Context IDs are system generated for each segment (dimension member) value, as shown in the following example.

Report Type	*Line Item*	*Entity*	*Segment*	*Period*	*Amount*	*Context ID*
Balance sheet	Disposals	ABC Corp		As of Dec31, 2008	100,000	AsofDec31
Disclosure - Plant Property & Equipment	Additions	ABC Corp	Building	As of Dec31, 2008	15,000	AsofDec31_Building
Disclosure - Plant Property & Equipment	Disposals	ABC Corp	Building	As of Dec31, 2008	20,000	AsofDec31_Building
Disclosure - Plant Property & Equipment	Disposals	ABC Corp	Land	As of Dec31, 2008	23,000	AsofDec31_Land

Uploading and Maintaining XBRL Facts and Taxonomy Elements

This section provides an overview of data upload requirements and discusses how to:

- Review file definitions.
- Upload instance facts and taxonomy elements.
- Maintain taxonomy elements.
- Maintain footnotes.
- Maintain instance facts.

Understanding Data Upload Requirements

We provide a generic data upload utility, the Load Data Application Engine process, to import your mapped taxonomy elements, financial instance facts, and disclosure facts into the system. File layouts are delivered for comma separated value (CSV) and tab separated value (TSV) text files. Typically, you will prepare the data using a spreadsheet program and save it as a CSV or TSV file. The following sections describe the required file layouts.

File Layout for Taxonomy Elements

The following table lists the required format for the file that contains the mapped taxonomy elements.

Field Nbr	*Field Name*	*Long Name*	*Field Type*	*Format*	*Length*
1	PS_ELEMENT	Element Id in PSFT	CHAR		30
2	EFFDT	Effective Date	DATE	YYYY/MM/DD	10
3	TAXONOMY_ID	Taxonomy ID	CHAR		10
4	NAMESPACE_ALIAS	Namespace Alias	CHAR		10
5	XBRL_ELEMENT_NAME	Element Name	CHAR		254
6	EFF_STATUS	Status as of Effective Date	CHAR		1
7	XBRL_ELEMENT_LBL	XBRL Element Label	CHAR		254
8	DRCR	XBRL Balance	CHAR		15

Field Nbr	Field Name	Long Name	Field Type	Format	Length
9	XBRL_DATATYPE	Data Type	CHAR	NUM/TXT	3

The following example shows a spreadsheet that contains taxonomy elements using the required format. PS Element is the user-defined ID you will use in PeopleSoft EPM for the element.

PS Element	Effective Date	Status	Taxonomy Id	NameSpace Alias	XBRL Element ID	XBRL Element Label	Balance	Data Type
PROVISION	2009/01/01	A	USGAAP	us-gaap	ProvisionForDoubtfulAccounts	Provision for Doubtful Accounts	debit	NUM
AR_CURR	2009/01/01	A	USGAAP	us-gaap	ReceivablesNetCurrent	Receivables, Net, Current	Debit	NUM
DEBT_LT_PAY	2009/01/01	A	USGAAP	us-gaap	RepaymentsOfLongTermDebt	Repayments of Long-term Debt	credit	NUM
EXP_R&D	2009/01/01	A	USGAAP	us-gaap	ResearchAndDevelopmentExpense	Research and Development Expense	Debit	NUM
RETAINED_EARNINGS	2009/01/01	A	USGAAP	us-gaap	RetainedEarningsAccumulatedDeficit	Retained Earnings (Accumulated Deficit)	Credit	NUM
REV_SALES	2009/01/01	A	USGAAP	us-gaap	SalesRevenueGoodsGross	Sales Revenue, Goods, Gross	Credit	NUM
REV_SERVICES	2009/01/01	A	USGAAP	us-gaap	SalesRevenueServicesNet	Sales Revenue, Services, Net	Credit	NUM
INVEST_SHORT	2009/01/01	A	USGAAP	us-gaap	ShortTermInvestments	Short-term Investments	Debit	NUM
REV_TECH	2009/01/01	A	USGAAP	us-gaap	TechnologyServicesRevenue	Technology Services Revenue	Credit	NUM
SEG_SUB_ARRANGE	2009/01/01	A	USGAAP	us-gaap	SubscriptionArrangementMember	Subscription Arrangement	none	TXT
T_REVREC_GP	2009/01/01	A	USGAAP	us-gaap	RevenueRecognitionGeneralPrinciples	Revenue Recognition, General Principles	none	TXT
T_REVREC_LP	2009/01/01	A	USGAAP	us-gaap	RevenueRecognitionLoyaltyPrograms	Revenue Recognition, Loyalty Programs	none	TXT
DIM_REV_DEF_ARRANGE	2009/01/01	A	USGAAP	us-gaap	DeferredRevenueArrangementTypeAxis	Deferred Revenue Arrangement Type [Axis]	none	TXT
SEG_LAYAWAY_SALE	2009/01/01	A	USGAAP	us-gaap	LayawaySaleMember	Layaway Sale	none	TXT

Taxonomy Elements

Note. The date field format is YYYY/MM/DD. For example, the date Jul 01, 2009 should be formatted as 2009/07/01.

File Layout for Financial Report Facts

The following table lists the required format for the file that contains the element facts for financial reports.

Field Nbr	Field Name	Edit Table	Long Name	Field Type	Length
1	TAXONOMY_ID	XBRL_TAXONOMY	Taxonomy ID	CHAR	10
2	ENTITY_NAME	XBRL_ENTITY	Entity Name	CHAR	50
3	REPORT_ID	XBRL_RPT_DEFN	Report ID	CHAR	8
4	XBRL_CONTEXT_ID	XBRL_CONTEXT	XBRL Context Identifier	CHAR	30
5	LINE_NBR		Line Number	NUM	5
6	PS_ELEMENT	XBRL_ELEMENTS	Element ID in PSFT	CHAR	30
7	XBRL_UOM_ID	XBRL_UOM	XBRL Unit of Measure ID	CHAR	30
8	XBRL_DECIMALS		Rounded to Decimals	CHAR	20

Field Nbr	Field Name	Edit Table	Long Name	Field Type	Length
9	XBRL_AMOUNT		Amount	CHAR	30
10	XBRL_FOOTNOTE_ID	XBRL_FOOTNOTES	Foot Note Id	CHAR	30

The following example shows a spreadsheet that contains element facts for financial reports using the required format. PS Element is the user-defined ID you will use in PeopleSoft EPM for the element.

Taxonomy Id	Entity Name	Report	Context Id	Line#	PS Element	UOM ID	Decimals	Amount	Footnote ID
USGAAP	ABC_CORP_SEC	BAL_SHT	AS_OF_JUNE30_2008	1	CASH	USD	-6	20000000	
USGAAP	ABC_CORP_SEC	BAL_SHT	AS_OF_JUNE30_2008	2	INVEST_SHORT	USD	-6	10000000	
USGAAP	ABC_CORP_SEC	BAL_SHT	AS_OF_JUNE30_2008	3	AR_CURR	USD	-6	5000000	
USGAAP	ABC_CORP_SEC	BAL_SHT	AS_OF_JUNE30_2008	4	EXP_PREPAID	USD	-6	3000000	
USGAAP	ABC_CORP_SEC	BAL_SHT	AS_OF_JUNE30_2008	5	ASSETS_OTHER_CURR	USD	-6	15000000	
USGAAP	ABC_CORP_SEC	BAL_SHT	AS_OF_JUNE30_2008	6	INVENTORY_NET	USD	-6	4000000	
USGAAP	ABC_CORP_SEC	BAL_SHT	AS_OF_JUNE30_2008	7	ASSETS_CURR	USD	-6	30000000	
USGAAP	ABC_CORP_SEC	BAL_SHT	AS_OF_JUNE30_2008	8	PPE_NET	USD	-6	8000000	PPE_JUN30_2008
USGAAP	ABC_CORP_SEC	BAL_SHT	AS_OF_JUNE30_2008	9	DEBT_CURR	USD	-6	4000000	
USGAAP	ABC_CORP_SEC	BAL_SHT	AS_OF_JUNE30_2008	10	AP_CURR	USD	-6	2000000	
USGAAP	ABC_CORP_SEC	BAL_SHT	AS_OF_JUNE30_2008	11	COMMON_STOCK	USD	-6	40000000	
USGAAP	ABC_CORP_SEC	BAL_SHT	AS_OF_JUNE30_2008	12	CAPITAL_OTHER	USD	-6	5000000	
USGAAP	ABC_CORP_SEC	BAL_SHT	AS_OF_JUNE30_2008	13	RETAINED_EARNINGS	USD	-6	9000000	
USGAAP	ABC_CORP_SEC	BAL_SHT	AS_OF_JUNE30_2008	14	T_REVREC_GP	USD	-6	000000	STD_REVREC_GP
USGAAP	ABC_CORP_SEC	BAL_SHT	AS_OF_JUNE30_2008	15	T_REVREC_LP	USD	-6	000000	STD_REVREC_LP
USGAAP	ABC_CORP_SEC	INCOME	FISCAL_YEAR_2008	1	INCOME_OPERATING	USD	-6	5000000	
USGAAP	ABC_CORP_SEC	INCOME	FISCAL_YEAR_2008	2	INCOME_OTHER	USD	-6	4000000	
USGAAP	ABC_CORP_SEC	INCOME	FISCAL_YEAR_2008	3	REV_SALES	USD	-6	20000000	
USGAAP	ABC_CORP_SEC	INCOME	FISCAL_YEAR_2008	4	REV_SERVICES	USD	-6	10000000	
USGAAP	ABC_CORP_SEC	INCOME	FISCAL_YEAR_2008	5	REV_TECH	USD	-6	3000000	
USGAAP	ABC_CORP_SEC	INCOME	FISCAL_YEAR_2008	6	REV_LICENSE	USD	-6	6000000	
USGAAP	ABC_CORP_SEC	INCOME	FISCAL_YEAR_2008	7	EXP_R&D	USD	-6	1000000	
USGAAP	ABC_CORP_SEC	INCOME	FISCAL_YEAR_2008	8	EXP_DEPR	USD	-6	1000000	
USGAAP	ABC_CORP_SEC	INCOME	FISCAL_YEAR_2008	9	EXP_OTHER	USD	-6	1000000	
USGAAP	ABC_CORP_SEC	INCOME	FISCAL_YEAR_2008	10	EPS_BASIC	EPS	0	50	

Element Facts for Financial Reports

The decimal column specifies rounding. If the amount is rounded to thousands, the decimal value is "-3". If the amount is rounded to millions, the decimal value is "-6". The Amount field should be populated as per the XBRL guidance.

You should ensure that these are not comma formatted. For Example, 2 million USD rounded to millions should be represented as:

Decimal = -6 and Amount = 2000000

File Layout for Disclosure Schedule

The disclosure schedule (table) has two additional columns for Dimension and Segment (dimension member). The following table lists the required format for the file that contains the disclosure facts.

Field Nbr	Field Name	Edit Table	Long Name	Field Type	Length
1	TAXONOMY_ID	XBRL_TAXONOMY	Taxonomy ID	CHAR	10
2	ENTITY_NAME	XBRL_ENTITY	Entity Name	CHAR	50
3	REPORT_ID	XBRL_RPT_DEFN	Report ID	CHAR	8
4	XBRL_CONTEXT_ID	XBRL_CONTEXT	XBRL Context Identifier	CHAR	30
5	LINE_NBR		Line Number	NUM	5
6	PS_ELEMENT	XBRL_ELEMENTS	Element ID in PSFT	CHAR	30
7	XBRL_DIMENSION	XBRL_ELEMENTS	Dimension	CHAR	30
8	XBRL_SEGMENT	XBRL_ELEMENTS	XBRL Segment	CHAR	100
9	XBRL_UOM_ID	XBRL_UOM	XBRL Unit of Measure ID	CHAR	30
10	XBRL_DECIMALS		Rounded to Decimals	CHAR	20
11	XBRL_AMOUNT		Amount	CHAR	30
12	XBRL_FOOTNOTE_ID	XBRL_FOOTNOTES	Foot Note ID	CHAR	30

The following example shows a spreadsheet that contains disclosure facts using the required format.

A	B	C	D	E	F	G	H	I	J	K	L
Taxonomy Id	Entity Name	Report	Context Id	Line#	PS Element ID	Dimension	XBRL Segment	UOM ID	Decimals	Amount	Footnote ID
USGAAP	ABC_CORP_SEC	DISC-DR	SALE_JUNE30_2008	1	REV_DEF_CURR	DIM_REV_DEF_ARRANGE	SEG_LAYAWAY_SALE	USD	0	60000	
USGAAP	ABC_CORP_SEC	DISC-DR	SALE_JUNE30_2008	2	REV_DEF_NCURR	DIM_REV_DEF_ARRANGE	SEG_LAYAWAY_SALE	USD	0	45000	
USGAAP	ABC_CORP_SEC	DISC-DR	SALE_JUNE30_2008	3	REV_DEF_CURR	DIM_REV_DEF_ARRANGE	SEG_SUB_ARRANGE	USD	0	50000	

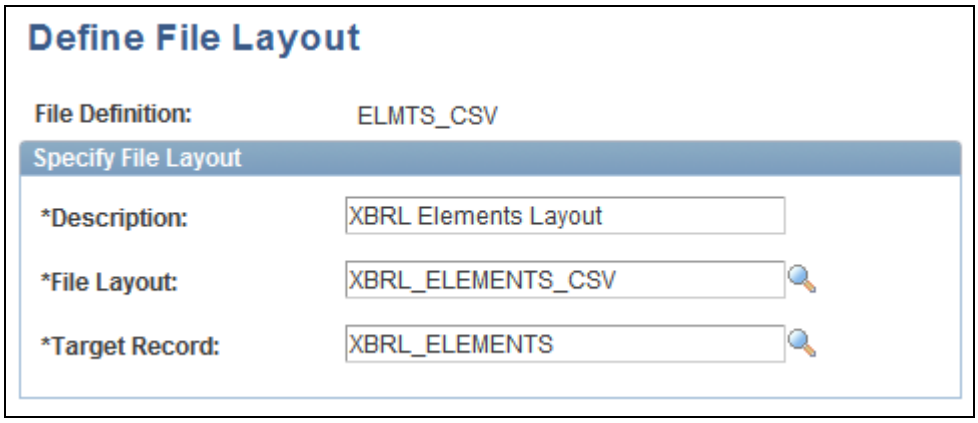
Disclosure Facts Formatting

Pages Used to Upload Facts and Taxonomy Elements

Page Name	*Definition Name*	*Navigation*	*Usage*
Define File Layout	XBRL_FILE_DFN	EPM Foundation, XBRL Reporting, Import Data, Define File Layout	Review the delivered file definitions and file layouts.
Load Data	XBRL_DATA_LD	EPM Foundation, XBRL Reporting, Import Data, Load Data	Upload taxonomy elements or instance facts.
Maintain Taxonomy Elements	XBRL_ELEMENTS	EPM Foundation, XBRL Reporting, Maintain Data, Maintain Taxonomy Elements	Review or update taxonomy elements that you have uploaded.
Maintain Footnotes	XBRL_FOOTNOTES	EPM Foundation, XBRL Reporting, Maintain Data, Maintain Footnotes	Review or update footnotes.
Maintain Footnotes – Fact References	XBRL_FNOTES_REF	EPM Foundation, XBRL Reporting, Maintain Data, Maintain Footnotes, Fact References	Review or update footnote fact references.
Maintain Instance Facts	XBRL_INST_FACT	EPM Foundation, XBRL Reporting, Maintain Data, Maintain Instance Facts	Review or update instance facts that you have uploaded.

Reviewing File Definitions

Access the Define File Layout page (EPM Foundation, XBRL Reporting, Import Data, Define File Layout).



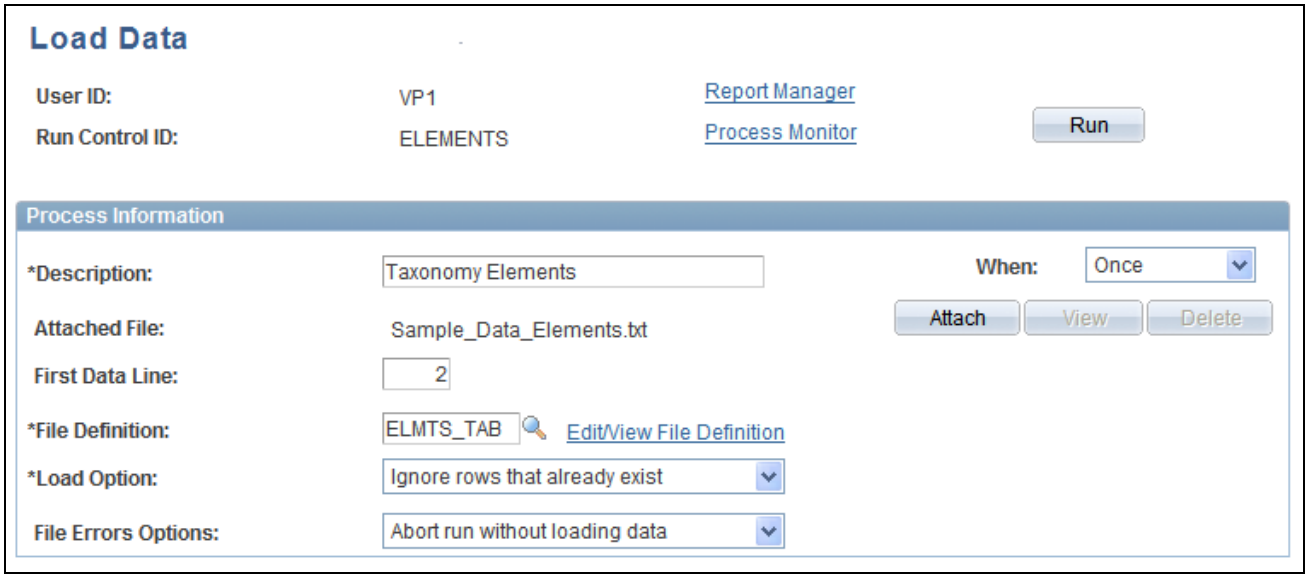
Define File Layout page

The following table lists the delivered file definitions. You should not modify these.

File Definition	File Layout	Format	Description
DISCL_CSV	XBRL_DISCL_FACTS_CSV	CSV	Use this file layout to upload disclosure schedule data in CSV format.
DISCL_TSV	XBRL_DISCL_FACTS_TSV	TSV	Use this file layout to upload disclosure schedule data in TSV format.
ELMTS_CSV	XBRL_ELEMENTS_CSV	CSV	Use this file layout to upload taxonomy elements in CSV format.
ELMTS_TSV	XBRL_ELEMENTS_TSV	TSV	Use this file layout to upload taxonomy elements in TSV format.
FACTS_CSV	XBRL_INST_FACT_CSV	CSV	Use this file layout to upload financial instance facts in CSV format.
FACTS_TSV	XBRL_INST_FACT_TSV	TSV	Use this file layout to upload financial instance facts in TSV format.

Uploading Instance Facts and Taxonomy Elements

Access the Load Data run control page (EPM Foundation, XBRL Reporting, Import Data, Load Data).



Load Data run control page

Attach Click this button to browse to and attach the file to import.

View Click this button to view an attached file.

Delete Click this button to delete an attached file.

First Data Line Enter the line number where the actual data begins. For example, if the first row of your file contains headings, then you should enter 2, if not, enter 1.

File Definition Specify the file definition to use for the import. The delivered file definitions that you use are:

- *DISCL_CSV:* Select to import disclosure facts in CSV format.
- *DISCL_TSV:* Select to import disclosure facts in TSV format.
- *ELMTS_CSV:* Select to import taxonomy elements in CSV format.
- *ELMTS_TSV:* Select to import taxonomy elements in TSV format.
- *FACTS_CSV:* Select to import financial element facts in CSV format.
- *FACTS_TSV:* Select to import financial element facts in TSV format.

Load Option Specify how to handle duplicate rows. Values are:

- *Insert and/or Update:* Select to load all rows. If duplicate rows exist, they are replaced with the new data in the source file.
- *Ignore rows that already exist:* Select to only load new rows. If duplicate rows exist they are rejected. In other words, the existing data is retained, while new data is loaded.

File Errors Option Specify how to handle errors during the import process. Value are:

- *Abort run without loading data:* Select to reject the file without loading any data. Error rows are written to the error log.
- *Skip errors rows and continue:* Select to load only valid rows. Rejected rows are written to the error log.

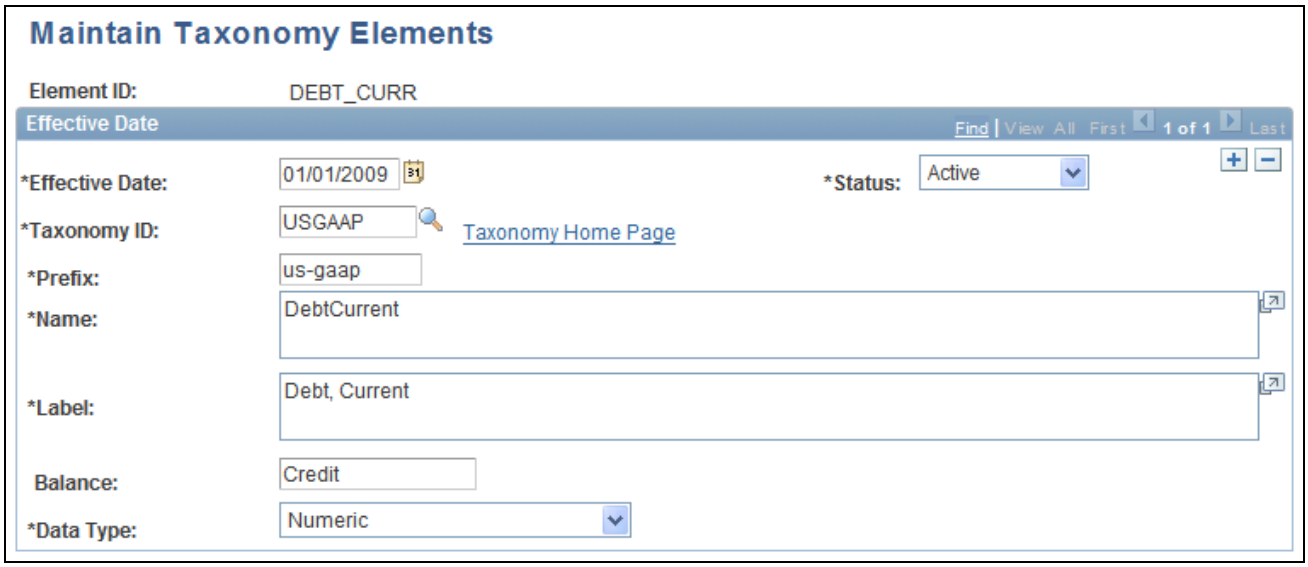
Run Click to start the data upload process.

After running the process, review errors logs (if any) and take the necessary corrective actions.

The Load Data process does not use EPM job streams. Instead, the process uses online temporary table instances, and Peopletools manages the assignment of temporary tables. The instance count for the Load Data process has been set to 3. Make sure that the "Online Temp Table Instances" parameter in PeopleTools Options is configured to at least 3.

Maintaining Taxonomy Elements

Access the Maintain Taxonomy Elements page (EPM Foundation, XBRL Reporting, Maintain Data, Maintain Taxonomy Elements).



Maintain Taxonomy Elements page

Use this page to maintain the taxonomy elements that you upload.

Element ID Each element ID represents a combination of taxonomy element name and the namespace it belongs to. The element ID is referenced in the instance facts, and also when you create an XBRL instance. This is a user-defined field, in other words, you decide what ID to use.

Taxonomy ID The taxonomy this element is associated with. You set up taxonomy IDs using the Define Taxonomy page.

Prefix The namespace prefix (alias) for the element.

Name The taxonomy element name, as defined in the taxonomy schema.

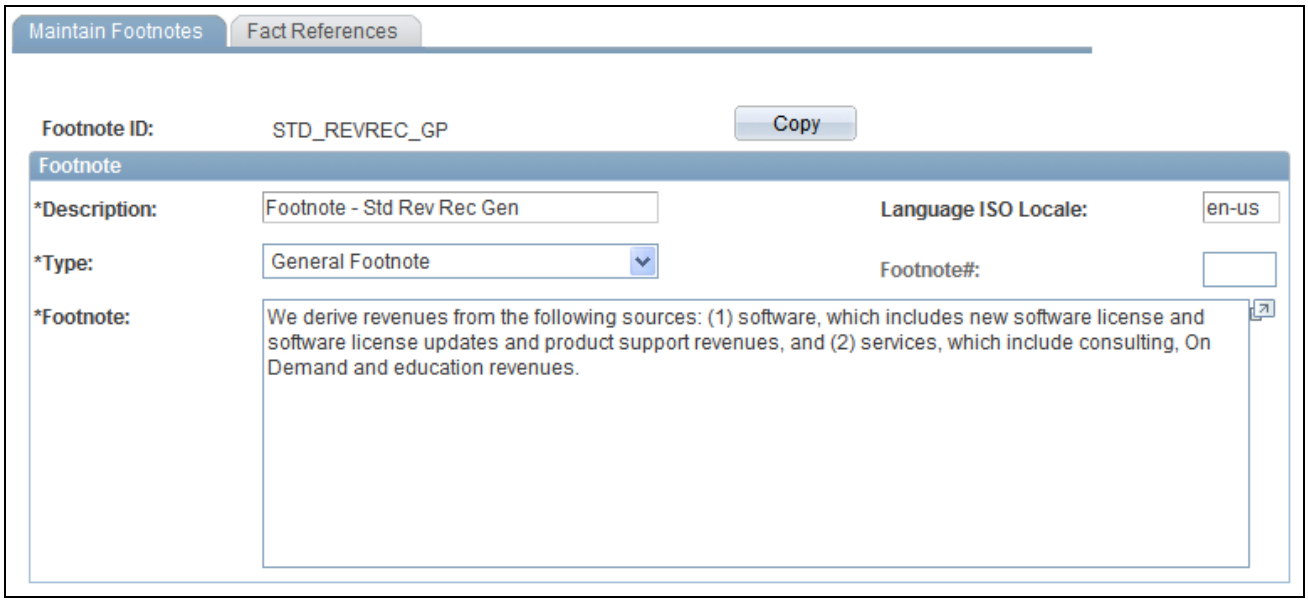
Label The taxonomy element label, as defined in the taxonomy schema.

Balance Balance type can be Debit or Credit. This field is blank for some elements that are used for footnotes and cash flow line items.

Data Type Values are *Numeric* or *Text*.

Maintaining Footnotes

Access the Maintain Footnotes page (EPM Foundation, XBRL Reporting, Maintain Data, Maintain Footnotes).



Maintain Footnotes page

You can create or update footnotes using this page, and use the Fact References tab to review the instance facts they are associated with. The footnotes may be associated to an element representing a report line or it may have its own element. In the later case, the amount and UOM fields will be blank.

Copy Click this button to copy the current footnote to a new Footnote ID.

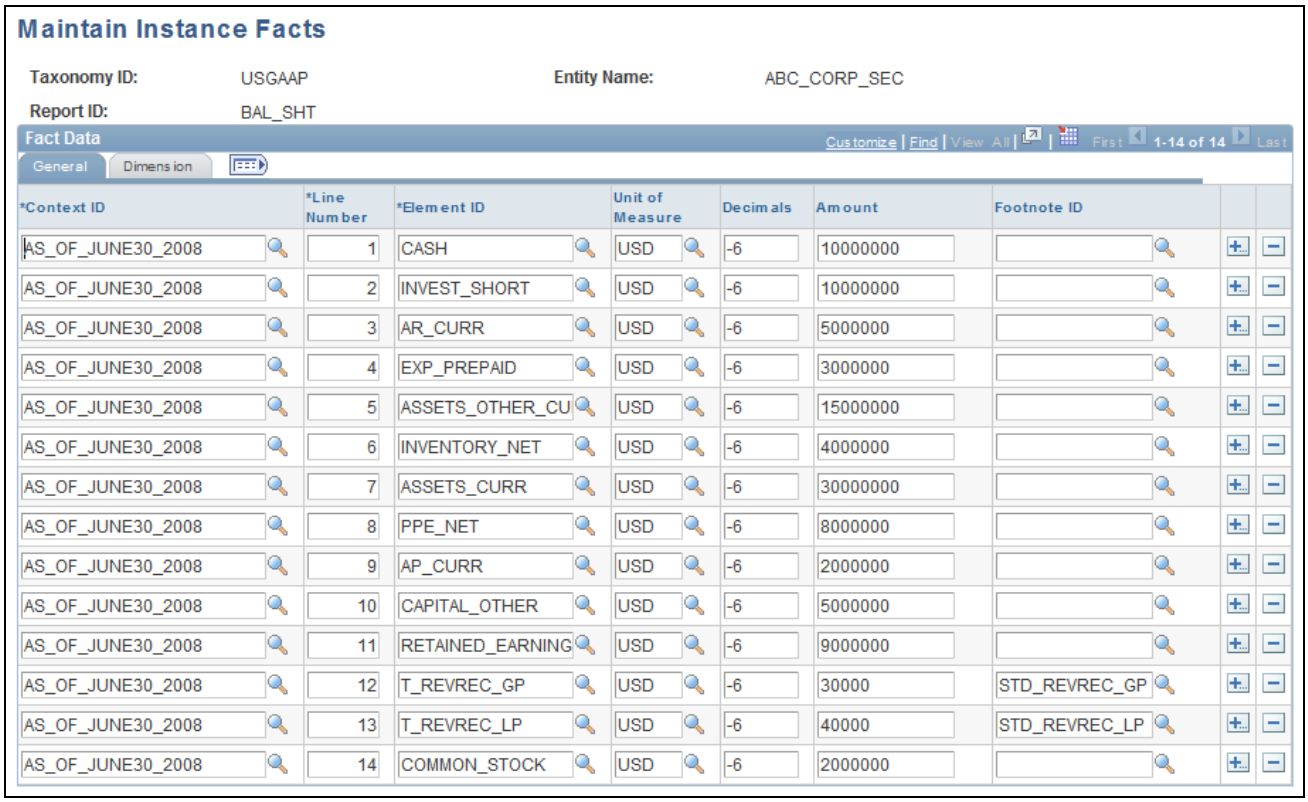
Language ISO Locale Enter the ISO language code for the footnote.

Type Specify the type of footnote. Values are:

- *General Footnote:* Select for a footnote that does not reference a report line item.
- *Reference to Report Line Item:* Select for a footnote that references a line item on the report, and enter the report line number it is associated with in the Footnote# field.

Maintaining Instance Facts

Access the Maintain Instance Facts page (EPM Foundation, XBRL Reporting, Maintain Data, Maintain Instance Facts).



Maintain Instance Facts

Taxonomy ID: USGAAP Entity Name: ABC_CORP_SEC

Report ID: BAL_SHT

Fact Data — Customize | Find | View All | First 1-14 of 14 Last

General | Dimension

*Context ID	*Line Number	*Element ID	Unit of Measure	Decimals	Amount	Footnote ID
AS_OF_JUNE30_2008	1	CASH	USD	-6	10000000	
AS_OF_JUNE30_2008	2	INVEST_SHORT	USD	-6	10000000	
AS_OF_JUNE30_2008	3	AR_CURR	USD	-6	5000000	
AS_OF_JUNE30_2008	4	EXP_PREPAID	USD	-6	3000000	
AS_OF_JUNE30_2008	5	ASSETS_OTHER_CU	USD	-6	15000000	
AS_OF_JUNE30_2008	6	INVENTORY_NET	USD	-6	4000000	
AS_OF_JUNE30_2008	7	ASSETS_CURR	USD	-6	30000000	
AS_OF_JUNE30_2008	8	PPE_NET	USD	-6	8000000	
AS_OF_JUNE30_2008	9	AP_CURR	USD	-6	2000000	
AS_OF_JUNE30_2008	10	CAPITAL_OTHER	USD	-6	5000000	
AS_OF_JUNE30_2008	11	RETAINED_EARNING	USD	-6	9000000	
AS_OF_JUNE30_2008	12	T_REVREC_GP	USD	-6	30000	STD_REVREC_GP
AS_OF_JUNE30_2008	13	T_REVREC_LP	USD	-6	40000	STD_REVREC_LP
AS_OF_JUNE30_2008	14	COMMON_STOCK	USD	-6	2000000	

Maintain Instance Facts page

Use this page to review and maintain the instance facts that you upload. You can associate footnotes with a fact by selecting a Footnote ID.

Creating XBRL Instance Documents

This section discusses how to create an XBRL instance document. Page Used to Create Instance Documents.

Page Used to Create an XBRL Instance Document

Page Name	*Definition Name*	*Navigation*	*Usage*
Create Instance run control	XBRL_INST_RUN	EPM Foundation, XBRL Reporting, Generate Instance, Create Instance	Generate an XBRL instance report.

Creating an XBRL Instance Document

After all of your setup is complete, and you've imported your taxonomy and financial facts, you can create an XBRL instance document using the Create Instance application engine process.

Access the Create Instance run control page (EPM Foundation, XBRL Reporting, Generate Instance, Create Instance).

Create Instance run control page

This page uses all of the set up information to create an XBRL instance report.

Instance Header Specify the instance header to use for this XBRL instance report. You establish an instance header using the Define Instance Header page.

As of Date Specify the date for the instance. This date is used to resolve the setup rules with effective dates.

Edit/View Header Click this link to access the Define Instance Header page, where you can review or modify the instance header definition.

Entity Name Select the entity that you will be submitting the XBRL instance document to. You establish XBRL entities using the Define Reporting Entity page.

Edit/View Entity Click this link to access the Define Reporting Entity page, where you can review or modify the reporting entity definition.

Report Output Specify the type of report output. Values are:

- *Report ID:* Select to generate an instance document from a single report ID, then select the report in the Report ID field.
- *Report Set:* Select to generate an instance document for a report set, then select the reports set in the Report Set field.

Preview Select this check box to create an XML publisher report for all the line items including footnotes. Use this report to verify the results of the XBRL instance document.

Note. XML Publisher must be installed for this option to work. For more information about XML Publisher.

See *PeopleSoft Enterprise PeopleTools PeopleBook: XML Publisher for PeopleSoft Enterprise*

Create Select this check box to create the XBRL instance document.

Publish If the Create option is selected, then you can also select this check box to create an XML message using Integration Broker. The message type is non-rowset based. You can view the Instance document as an XML message using the Service Operations Monitor of the Integration Broker.

Note. You need to set up Integration Broker for this option to work. Only the message object is delivered.

See *PeopleSoft Enterprise PeopleTools PeopleBook: PeopleSoft Integration Broker Administration*

File Name Enter the filename to assign to the generated XBRL instance report, without the file extension. The system automatically appends ".XBRL" and ".XML" to the file name.

The content of the XML and XBRL files are identical. You will need to submit the XBRL file per the XBRL requirements.

The Create Instance process does not use EPM job streams. Instead, the process uses online temporary table instances, and Peopletools manages the assignment of temporary tables. The instance count for the Create Instance process has been set to 3. Make sure that the "Online Temp Table Instances" parameter in PeopleTools Options is configured to at least 3.

To confirm the document is well formed, open it in an XML editor or an Internet browser, such as Internet Explorer or Mozilla Firefox. If it loads without errors, it is well-formed.

After the process is complete, review the process monitor status and the message logs for any logged exceptions. You can access the instance documents using the process monitor or the report manager if the report node is setup properly.

The context IDs are system generated for each segment (dimension member) value of a table schedule.

Appendix A

ETL Installation and Implementation Prerequisites and Considerations

This appendix provides initial ETL preparation tasks and discusses how to:

- Size your database and DataStage servers.
- Determine server configuration and install IBM WebSphere DataStage.
- Use IBM WebSphere DataStage implementation considerations.
- Define a job execution strategy.

See Also

Chapter 7, "Preparing to Load Source Data Into EPM," page 161

Chapter 11, "Setting Up DataStage for EPM," page 255

Preparing to Install and Implement IBM WebSphere DataStage

Perform the following preparatory tasks before you begin implementing ETL jobs:

1. Create a detailed list of all the EPM products that have been purchased and the related license codes.

   Identify and enumerate the products you are going to implement and in what order.

2. Create a detailed implementation schedule that accounts for the EPM data marts and business units you are going to implement.

3. Review the list of ETL application software components (such as .dsx, parameter, and DSParams files) and identify which are necessary for your requirements based on your implementation schedule.

   See Appendix C, "ETL Reference Documents," page 671.

4. Identify the list of database tables that will be populated and the list of corresponding jobs that have to be executed to populate these tables.

   Note. Apart from the jobs, which directly populate the relevant target tables, you must also identify all the dependent jobs, such as hash file load jobs.

5. Perform all non-ETL implementation tasks.

Sizing Your Database/DataStage Servers

EPM uses hash files extensively, which are stored in a directory which you specify on the server. It is important, therefore, to remember this when you are determining sizing requirements because the amount of data in the hash files will increase with time. Also the server directory should hold the flat files as well as XML file inputs that the ETL process requires. Generally, every staging table has a corresponding hash file, and every dimension table has a corresponding hash file, so the size of all the hash files is a function of the size of the data that is stored in staging tables and the dimension tables. However, it is also to remember that only relevant columns in a table are loaded into a hash file.

For sizing the space requirement for hash files, we suggest that you take a few sample hash files and compare them with the underlying tables to determine the size requirement. Also compare the structure of the table and the number of columns in that table that are actually loaded to the hash file. It is very important to keep sufficient buffer size for future incremental data, since as the data size increases with time the hash files also grow in size. Another way to do this is with the help of an unsupported tool provided along with the IBM WebSphere DataStage CD. The tool is called HFC.exe, which is short for Hash File Calculator.

Perform the following server sizing tasks before you begin implementing ETL jobs:

1. Refer to all relevant database sizing documents delivered with EPM, and thoroughly familiarize yourself with it before implementation.

2. Perform database sizing, considering all the tables that are populated by the ETL process as well as those used for reporting.

3. Run the delivered script for inserting a *Not Available* row into all relevant tables.

   This script will insert one Not Available row each into every table, which is a prerequisite for the ETL application.

   Note. You can find the script on the installation CD in the following location: <PSHOME>\SRC\ETL.

   ---

4. To size the DataStage server, determine the number of hash files that will be created for the subset of the ETL application that you are going to implement.

   You can use the list of jobs you have created in previous steps and the list of hash files that are supplied along with EPM.

5. Calculate the space required for storing all of these hash files.

   You must consider hash file properties and structure, as well as the quantum of data that is associated to each hash file to perform hash file sizing.

   Note. A buffer should be allocated for future incremental data (growth in the size of the hash file).

   ---

6. Decide where you will physically store hash files by setting the value in the environmental parameter.

   Space is also required for Datastage server log files.

7. Allocate space for all the other input data files such as XML files, parameter files, and *.dat files.

DataStage Server Requirements

Please see the *IBM Information Server: Planning Installation and Configuration Guide* for the minimum requirements to install the DataStge Server on a specific platform.

DataStage Client Requirements

Please see the *IBM Information Server: Planning Installation and Configuration Guide* for the minimum requirements to install the DataStge Client.

Determining Server Configuration and Installing IBM WebSphere DataStage

Perform the following server configuration and installation tasks before you begin implementing ETL jobs:

1. Determine a suitable server configuration for your development, QA, and production environments.
2. Install the DataStage servers.

   Create separate servers for development, QA, and production environments.
3. Perform all required steps to configure the database, depending on your source and target databases.
4. Install the DataStage client.
5. Apply the latest patches for DataStage server and client.

IBM WebSphere DataStage Implementation Considerations

The following considerations should be noted before you begin DataStage implementation:

1. Perform a detailed analysis of your project creation strategy.

   You should decide whether you would like a single project for the whole EPM application or have separate projects for each data mart.
2. Create separate DataStage projects for development, QA, and production.

   PeopleSoft recommends that the production project reside on a separate DataStage server.
3. Classify your jobs as high, medium, and low volume.

   Provide project defaults for array size, transaction size, IPC buffer and other performance parameters. Any exceptions and special cases must be handled by changing the value at the job level.
4. Open a sample job from each category and familiarize yourself with the filter conditions in the source, update strategy, job design, job parameters and other transformations.

5. Review the master run utility and create appropriate sequential file inputs.

   Analyze this feature and decide on the different categories that you want to run using this utility.

6. Review the master sequencers and familiarize yourself with them.

7. Open one of the business process and identify all the jobs that are required to run it.

   Run this as an example to learn how the jobs are ordered, the interdependencies, the hash file usage, and so forth.

Defining a Job Execution Strategy

The following job execution strategies should be noted before you begin running jobs:

1. Plan a job scheduling strategy and use the DataStage Director scheduler or another third-party tool.

   Do a sample run using the scheduling tool to test whether the tool meets all your requirements for scheduling the application.

2. Familiarize yourself with all the job execution utilities that are provided with DataStage.

3. Define the error validation strategy you wish to use in your job.

Appendix B

ETL Frequently Asked Questions

This appendix provides answers to frequently asked ETL questions for EPM, and covers these topics:

- IBM WebSphere DataStage ETL Tools
- DataStage Server Installation
- DataStage Client Installation
- DataStage Patches
- ETL Content
- Environmental Variables
- Project Creation, Import, and Export
- Setting Up Development and Production Environments
- ETL Jobs
- Survey Jobs
- ETL Hashed Files
- ETL Routines
- ETL Job Process Flow
- ETL Utilities
- Running ETL Jobs
- Error Handling with ETL Jobs
- ETL Job Validation
- Common ETL Issues
- Configuring Delivered ETL Content

IBM WebSphere DataStage ETL Tools

This table provides answers to general questions about IBM WebSphere DataStage ETL tools.

Question	*Answer*
What versions of IBM WebSphere DataStage are supported in EPM?	EPM only supports IBM WebSphere DataStage version 8.1 server edition.
What if I am using an earlier version of DataStage either as part of a non-PeopleSoft installation or as part of EPM?	You must upgrade to IBM WebSphere DataStage version 8.1 server edition
What are the differences between ETL content for EPM release 9.1 and the previous EPM releases?	For 9.1 OWS staging tables contain all the fields of its related source tables.
What are the IBM WebSphere software components that are delivered with EPM?	EPM delivers IBM InfoSphere Information Server 8.1
What is QualityStage?	IBM WebSphere QualityStage provides a powerful framework for developing and deploying data investigation, standardization, enrichment, probabilistic matching and survivorship operations. For use in transactional, operational, or analytical applications, in batch and real-time, the same services are seamlessly deployed to facilitate data validation, cleansing or master data entity consolidation for your, locations and products. For more information, please use the following link to the IBM WebSphere website: http://www-01.ibm.com/software/data/infosphere/qualitystage/
What is InfoSphere Information Analyzer?	For more information, please use the following link to the IBM WebSphere website: http://www-01.ibm.com/software/data/infosphere/information-analyzer/
What is IBM Metadata Workbench?	IBM Metadata Workbench provides Web-based exploration of information assets that are generated and used by IBM Information Server applications. IBM Metadata Workbench gives you out-of-the-box reporting on data movement, data lineage, and the impact of changes and dependencies. With the workbench, you can trace the data lineage of business intelligence reports to provide a basis for compliance with regulations such as Sarbanes-Oxley and Basel II. IBM Information Server components generate design time and runtime metadata, automatically storing that metadata in the IBM Information
What is Parallel Extender?	DataStage Parallel Extender (DS-PX) is a highly scalable parallel processing infrastructure package for the development and execution of data integration, data warehousing, business intelligence and analytical applications.
How does DataStage ensure application security?	Password and role based security can be effectively implemented in DataStage at a project level from the DataStage administrator.

Question	Answer
How does DataStage ensure data security?	EPM accesses data on databases using the DRS stage. The user ID and password for accessing the databases are parameterized as environmental variables and the password parameter can be set as an encrypted field in the DataStage Administrator. This ensures data security by restricting the database access passwords.

DataStage Server Installation

This table provides answers to DataStage Server installation questions.

Question	Answer
What are the prerequisites to the DataStage server installation?	For information on prerequisites to DataStage server installation, please refer to the *PeopleSoft Enterprise Performance Management Installation Guide.* See *IBM Information Server: Planning Installation and Configuration Guide*
What steps must I perform to install IBM WebSphere DataStage Server?	For information on prerequisites to DataStage server installation, please refer to the *PeopleSoft Enterprise Performance Management Installation Guide.* See *IBM Information Server: Planning Installation and Configuration Guide*
How do I verify if I have successfully installed the IBM WebSphere DataStage Server?	After you install and configure IBM InfoSphere DataStage and QualityStage, test the installation by logging in to the Administrator and Designer clients. See *PeopleSoft Enterprise Performance Management Installation Guide.*

DataStage Client Installation

This table provides answers to DataStage Client installation questions.

Question	Answer
How do I verify whether I have successfully installed DataStage client?	Please refer to the *PeopleSoft Enterprise Performance Management Installation Guide.*

DataStage Patches

This table provides answers to questions about DataStage patches.

Question	*Answer*
Are there any other relevant patches that I have to apply other than DataStage patches?	There are two patches for SQL Server and DB2 database users. These are DRS patches for these two databases. You can find these patches in the following resolutions: • 829720 - DRS patch for SQL server database users • 829719 - DRS patch for DB2 database users
How do I verify that the patch has been installed correctly?	Verify that the files specified in the install guide for the patch are updated properly to verify that the patches were installed correctly. Also, test some jobs in the project and ensure that no abnormal termination or any other errors occurs right after applying the patch.

ETL Content

This table provides answers to ETL content packaging questions.

Question	*Answer*
How is the prepackaged ETL content delivered to me?	The DataStage application is delivered in the form of **.dsx* files, which are text files that can contain export of a whole DataStage project or can contain DataStage design components and executables, like server jobs, sequencers, routines, containers and so forth.
Where can I find the list of delivered DSX files?	See Appendix C, "ETL Reference Documents," DSX File Import Descriptions, page 671.
How are the delivered dsx files packaged in the CD (and future bundles)?	There are 38 dsx files that are delivered with EPM. You can see dsx files for five warehouses and 1 for Common where you have all the dsx files that are common across the warehouses.
What are the non-DSX file deliverables with the CD (and future bundles)?	Other than DSX files, which contain DataStage job designs, the other deliverables are flat file inputs, xml file inputs and parameter files. See Appendix C, "ETL Reference Documents," Parameter and Source Data Files Information, page 671.

Question	*Answer*
What are the various EPM bundle files and how will they be packaged in the bundle1/bundle2/ICE Resolution?	Refer to the bundle posting related information from You Connection.
What are the various source application release versions that have been used for EPM 9.1 release?	• FSCM Source Release – FSCM 9.1 • CRM Source Release - CRM 9.1 • HRMS Source Release - HCM 9.1 • Campus Solutions Source Release -CS 9.0

Environmental Variables

This table provides answers to questions about environmental variables.

Question	*Answer*
What are environmental variables and how are they used?	Environment variables enable you to set the project defaults and these are used in jobs.
What is the DSParams file?	For any datastge project, all project levels, user defined environmental items are stored in a file called DSParams. It is located in <dshome>\DataStage\Projects\<projname>.

Question	***Answer***
How do I copy the DSParams file the very first time as opposed to copying from one project to another project?	The methodology described below provides a workaround for moving or sharing the global parameters without having to re-type them in the administrator. The work around consists of replacing and/or editing this file to add the parameters. Be sure to back up the original file before any other activity occurs. For a *new project* that has not yet defined any global parameters, just copy the existing DSParams file to the new project. Be sure to rename the existing DSParams file. Ensure all DS clients (Designers, etc.) are logged off, and stop and start the DataStage services to activate it. Then go into the DataStage Administrator and all the parameters should be visible in the user-defined section of the environment screen. At this point, edit the default values for each parameter. For an *existing project* that has already defined some global parameters; the DSParams file must be edited to add the desired parameters. The process below describes how to do this. The user-defined parameters are in two sections of the DSParams file: one section defines the parameters [EnvVarDefns] and the second section contains the default values [EnvVarValues]. The approach is to copy the correct lines from the original source project file into the target project DSParams file: 1. Rename/Backup the DSParams file in the target project directory and backup the source project DSParams file as well. 2. Edit the source project DSParams. Go to the end of the [EnvVarDefns] section and find the user defined parameters, which are at the end of the section. Select the lines up to but not including the line which contains [PROJECT]". 3. Copy these lines and paste them into the target project DSParams file before the "[PROJECT]" section. 4. Go back and edit the source project DSParams file. Find the section starting with the line "[EnvVarValues]". This is usually at the end of the file. Copy all of the lines of that section, or select all the lines for the specific parameters to be moved. 5. Locate the end of the DSParams file in the target project directory. See if it has a section called "[EnvVarValues]". If it does not, add it. If it does, then go to the next step. 6. Paste the lines into the target project DSParams file at the end of the "[EnvVarValues]" section and before the end of file. 7. Save the DSParams file in the target project directory.

Question	*Answer*
	8. Ensure all DS clients (designers and so forth) are logged off, and stop and start the DataStage services to activate it. Then go into the DataStage Administrator and all the parameters should be visible in the user-defined section of the environment screen. At this point, change the default values for each parameter.
What are 'Array' and 'Transaction' sizes?	• *Array Size:* Array size is a parameter to specify the number of rows written (to a database) at a time. In other words it refers to the number of rows that are transferred in one call between DataStage and the database before they are written. Generally, increasing the array size will increase performance since client memory is used to cache records resulting in lesser server hits. The maximum size for array size is 32767. But increasing the array size too much will result in strain on the client memory. Hence an optimal value must be arrived at considering the client memory. For flexibility this has been parameterized as a environmental variable. Separate environmental variables are available for each source as well as for OWS, OWE and MDW. For direct DRS lookups, the recommended array size is 1 since DS lookups generally expects 1 row of data to be returned. • *Transaction Size:* Transaction size refers to the number of rows that are written to the database before the data is committed. Giving a transaction size of zero will ensure that commit doesn't happen until all the records are written. The default value is 0. If the transaction size is set to 100 then the database table commits are performed every 100 rows. Here again an optimal value must be arrived at considering the strain on the Database server and the number of records. For flexibility this has also been parameterized as a environmental variable. Separate environmental variables are available for each source as well as for the OWS, OWE and MDW.
How is the DATA_ORIGIN environment variable configured?	The value for DATA_ORIGIN is 'S' for following loads: • OWS (data flow from PeopleSoft source system) • OWE (data flow from OWS) • MDW (data flow from OWS) The DATA_ORIGIN must be toggled to *Enterprise* only when running those sets of jobs whose flow is from OWE to MDW.

Project Creation, Import, and Export

This table provides answers to project creation, import, and export questions.

Question	*Answer*
What are the different methods of maintaining projects if I have jobs that source from more than one warehouse?	The EPM ETL design enables jobs to distribute across multiple projects or only a single project. However, the following are some of the options to create projects based on your needs. • *Option 1: One project per EPM warehouse* If you have more than one EPM warehouse, you can create one project per warehouse and one project for all the jobs related to Common jobs and Global Dimensions. For example, if you have HCM and FMS warehouse then there are three projects created for HCM, FMS and Common. The J_Dim_PS_D_PERSON is a part of Global Dimensions and this job must be triggered in Common project whereas the corresponding Hash Load jobs have to be triggered in HCM and FMS projects as well. • *Option 2: All EPM warehouses in a single project* You can have one single project for all the jobs for all the EPM warehouses, Common jobs and Global Dimensions. If you maintain only one project for all the warehouses (HCM, FMS and Common jobs) you need to run the Global Dimensions, Local Dimensions (HCM, FMS), SKU based dimensions and there is no need to run the Hash Load jobs after the first run. For this option, there is a possibility of reaching a maximum number of files limit for directories in some OS platforms. Please verify with system administrators before proceeding with this option.
What configurations must I perform after creating the projects?	Projects must be created from the DataStage Administrator. See Chapter 11, "Setting Up DataStage for EPM," Setting Up DataStage Projects, page 255.
How do I import a project? How do I verify a successful import?	For more information about importing a project, see section 2-5 of the Configuring IBM WebSphereDataStage document. See Chapter 11, "Setting Up DataStage for EPM," Setting Up DataStage Projects, page 255.
What are the DataStage categories (folders) and sub-categories that I will see after project import?	For more information about the DataStage categories available, see the topic 'Understanding the Project Structure' in this document.

Setting Up Development and Production Environments

This table provides answers to questions about development and production environment setup.

Question	***Answer***
Where can I find configuration information for different server platforms?	See *IBM Information Server: Planning Installation and Configuration Guide*

Question	*Answer*
What are the supported platforms for Data stage?	The following platforms support the IBM InfoSphere Information Server: • HP-UX 11i v2, v3 on PA-RISC • HP-UX 11i v2, v3 on Intel Itanium • IBM AIX 5.3 and 6.1 • Red Hat Enterprise Linux Advanced Server 4 on AMD or Intel processors • Red Hat Enterprise Linux 5 Advanced Platform on AMD or Intel processors • Red Hat Enterprise Linux 5 Advanced Platform on IBM System z • SUSE Linux Enterprise Server 10 on AMD or Intel processors • SUSE Linux Enterprise Server 10 on IBM System z • Sun Solaris 9 and Sun Solaris 10 • Microsoft Windows XP Service Pack 2, Microsoft Windows Vista, Microsoft Windows Server 2003 Service Pack 2 (Client only) • Microsoft Windows 2003 Service Pack 2 (Server only) Data sources Supported by EPM 9.1 application: • Oracle • MSSQL Server • IBM DB2 on OS/390 • IBM DB2 on Unix/Windows • Flat files (Sequential Files) • XML files Refer to the PeopleSoft Enterprise Performance Management Hardware and Software Requirements Guide for more details.

Question	*Answer*
Are there database-specific configuration steps?	For MSSQL Server, you must enable the options to support functional index (MSCONCATCOL) in the database: • SET ANSI_NULLS • QUOTED_IDENTIFIER • CONCAT_NULL_YIELDS_NULL • ANSI_WARNINGS • ANSI_PADDING

ETL Jobs

This table provides answers to general questions about ETL jobs.

Question	*Answer*
What are the different categories of jobs in EPM?	The following is a list of EPM job categories: • Source to OWS (Staging area) • OWS to OWE • OWS to MDW • OWE to MDW • OWS to MDW (for CRM online marketing data mart) • OWE to source (for GL and Position Budgeting in HCM)
What types of staging loads does EPM support?	EPM supports incremental staging loads along with a small number of destructive staging loads. See Chapter 7, "Preparing to Load Source Data Into EPM," Understanding ETL Load Strategies in EPM, page 176.
What is the load strategy for the MDW and OWE?	Most of the MDW and OWE loads are provided with Incremental logic using LASTUPD_EW_DTTM field which is part of the subrecord LOAD_OWS_SBR. See Chapter 7, "Preparing to Load Source Data Into EPM," Understanding ETL Load Strategies in EPM, page 176.

Question	*Answer*
How does EPM manage incremental loading if the source fields have Null values for the Datetime stamp?	If the Datetime column is a nullable field on the source database, then source filter will include a condition to bring that data as well along with the incremental data. See Chapter 7, "Preparing to Load Source Data Into EPM," Understanding ETL Load Strategies in EPM, page 176.
Are there any special loading requirements between the first run and subsequent runs?	No, there are no special loading requirements between the first run and subsequent runs, provided the prepackaged jobs are used without any customization.
How does EPM manage deletes in the source tables?	For the Campus Solutions Warehouse, EPM provides staging jobs that can identify source records that have been physically deleted from your PeopleSoft transaction system and flag those records for physical deletion from the Campus Solutions Warehouse schema. For more information see the chapter *Managing Source System Deletes and Archiving* in the *PeopleSoft Enterprise Campus Solutions Warehouse PeopleBook.*
Does EPM have the functionality to accommodate late arriving dimensions?	No, EPM does not have the functionality to accommodate late arriving dimensions.
What types of data sources does EPM accommodate?	EPM can accommodate XML Files, Flat Files, and RDBMS tables.
Does EPM use any control tables in its ETL design?	EPM no longer uses control tables and those that were used in past EPM releases (such as ETL_JOB_CTL, ETL_RUN_HST, ETL_TSE_ERR) have been deprecated.
Does EPM use surrogate keys? If so, how are Surrogate IDs generated and managed?	Datastage manages the universe file SDKSequences, which will hold the surrogate key sequences for a particular key. See Chapter 2, "Understanding PeopleSoft Enterprise Performance Management," Surrogate Keys, page 21.
Does EPM have a strategy for handling slowly changing dimensions?	In EPM, the dimension D_EMPL_JOB from HCM warehouse is designed as Type 2 slowly changing dimension and all the other dimension loads are Type 1. However, the lookup operation supports Type 2 slowly changing dimension (for example, whenever there is lookup on other dimension, it will have effective dated logic). For more information on configuring Type 2 slowly changing dimensions, see the chapter *Configuring Slowly Changing Dimensions* in your warehouse specific PeopleBook (for example, the *PeopleSoft Enterprise Campus Solutions Warehouse PeopleBook*).

Question	*Answer*
How to change my dimension load from Type 1 design to Type 2 design?	For more information on configuring Type 2 slowly changing dimensions, see the chapter *Configuring Slowly Changing Dimensions* in your warehouse specific PeopleBook (for example, the *PeopleSoft Enterprise Campus Solutions Warehouse PeopleBook*).
Does EPM support ETL rollbacks? If so, how is this handled?	Rollback is possible through the Transaction Size parameter. If the transaction size is selected as zero and if the job aborts in the middle, then the job will rollback the transactions since it follows the principle of two-way commit. If the transaction size is anything other than zero and if the job fails in the middle, then the job will perform commits for the number of rows that processed till the error message.
What aggregation strategy does EPM ETL employ?	The aggregator stage is generally not used in job design since the aggregation functions are better left to the database since the database can perform aggregation functions more efficiently than Datastage. Whenever the aggregation must be performed on the source data, it is achieved within DRS source stage itself. In case of generated sql queries, aggregate functions are given in against columns in corresponding derivation columns and group by clause is given in 'Other clauses' text area. Wherever User Defined SQL option is selected the query is specified appropriately with the aggregate function. In specific instances where an aggregation function must be performed on data that is transformed and not directly read from the Database and in. cases where the number of records is going to be large, temporary table is created where the data is temporarily written and then read out, when the aggregation functions can be performed.
What indexing strategy does EPM ETL employ?	EPM data models are delivered with indexes. Before loading the target tables, drop the indexes and then build them after load. This improves ETL performance.
How are lookups used in the EPM ETL design?	Lookups are usually used in a Hashed file stage, except for relational joins, when they are used in the DRS stage instead. See Chapter 7, "Preparing to Load Source Data Into EPM," Hashed Files, page 171 and Chapter 7, "Preparing to Load Source Data Into EPM," Understanding Data Validation and Error Handling in the ETL Process, page 179.

Question	*Answer*
What types of job parameters does EPM use to increase run time flexibility?	Parameterization helps you enter run time parameters without resorting to changing jobs. Run time information, such as the Database type, the database connection parameters, and parameter file directories should be set as environmental variables, which are used in individual jobs. Parameter files are used for those jobs, which read from the user, input variable values or a list of values, which may change from run to run. The variables and their respective values are given in parameter files. See Appendix C, "ETL Reference Documents," Parameter and Source Data Files Information, page 671.
How is DataStage code re-used?	PeopleSoft packages reusable DataStage code with Shared Containers, routines, and some server jobs (found in the Reusable Jobs folder).
Are there any customizations required to handle Unicode data?	To support Unicode databases, the DataStage Server must be installed with NLS enabled. Also, the proper character set should be selected based on the requirements by the user, in the DataStage Administrator.

Survey Jobs

This table provides answers to questions about survey jobs.

Question	***Answer***
Where are the Survey-related jobs in the dsx files? After import, where can I find the Survey jobs in the DataStage Project?	Survey jobs are present in OWE and MDW modules of HCM warehouse. In OWE module, there are some D00 jobs which reads the flat file data as source and loads the R00 tables. These jobs can be located in OWE.dsx and after the import, the jobs will be present under the OWE_E\HCM\D00\Base\Load_Tables\Server category. In the MDW module, the R00 tables are used as source and it load the Survey Dimension tables. These jobs can be located in WHR_WORKFORCE_PROFILE_MART_E.dsx file and after the import, the jobs will be present under the \HCM_E\WORKFORCE_PROFILE_MART\Survey\OWE_To_MDW\Dimensions\Base\Load_Tables\Server category. In EPM, the dimension D_EMPL_JOB from HCM warehouse is designed as a type 2 slowly changing dimension and all the other dimension loads are Type 1. However, the lookup operation supports Type 2 (for example, whenever there is lookup on other dimension, it will have effective dated logic).
What are the required steps in a Survey load?	1. *Run Survey Setup ETL:* These are the jobs that read the source flat files or the temp tables and loads the R00/D00 tables. These jobs can be located in OWE_E.dsx files and it will be present under the path \OWE_E\HCM\D00\Base\Load_Tables\Server category. 2. *Run Survey Load ETL Batch:* These are the jobs that read the data loaded in the above step and loads the F00 tables. These jobs can be located in OWE.dsx files and it will be present under the path \OWE_E\HCM\F00\Base\Load_Tables\Server category. 3. *Load Employee level Competencies:* These jobs loads the Competency details of the employee from the OWS tables. These jobs can be located in OWE.dsx files and it will be present under the path \OWE_E\HCM\F00\Base\Load_Tables\Server category.
What are the prerequisites for loading the Survey module?	OWE Survey jobs are present under the HCM warehouse. These jobs use Flat files as sources. These source flat files should be present in the Project home directory and the environmental parameter $SOURCE_FILE_DIR should have the directory path of these survey flat files. If the DataStage server is on Windows, then the survey jobs have to be modified by accessing the Sequential file stage and changing the Line Termination to *DOS Style (CR LF).* And then save the job, Compile and run the same.

ETL Hashed Files

This table provides answers to questions about ETL hashed files.

Question	*Answer*
How are hash files used and for what purpose?	Hash Files are used to enhance the performance of the ETL job. Hash Files are typically used for lookups in an ETL job. In EPM, there are jobs to initialize Hash Files. These jobs create the hash files before the jobs requiring them for lookup are executed. These Hash Files are also updated once the target table is loaded in the ETL job. This method will enable multiple jobs to utilize the same hash file as long as the structures required are the same. Another method is to load the hash file within the same job using them as a lookup. This method requires the hash files to be reloaded every time the job executes. See Chapter 7, "Preparing to Load Source Data Into EPM," Hashed Files, page 171 and Chapter 7, "Preparing to Load Source Data Into EPM," Understanding Data Validation and Error Handling in the ETL Process, page 179.
What should I keep in mind when managing my hash files?	The default setting for Hashed Files are project specific and cannot be shared across projects. The validity of Hashed Files is dependent on the base table it is generated from. The base table should only be updated by the ETL jobs provided in EPM. If not, the hashed file and the table will be out of sync and may result in faulty data when used in an ETL job. There are several Hashed File utilities provided in EPM. These are located in the Utilities\Hash_Utils category.
Can I customize the storage location for hash files?	It is possible to customize the storage location for hash files by specifying the directory path. You can set the storage path of the hash files. The path location has to be set in the environmental parameter #$HASHED_FILE_DIRECTORY# and this parameter is used across all the hash files.
How to recover data from corrupted hash files?	Generally, a corrupted hash file must be reloaded from the base table. EPM provides utilities to back up and recover DateTime and SurrogateKey hashed files.

ETL Routines

This table provides answers to questions about ETL routines.

Question	*Answer*
How are routines used?	Routines are used to make DataStage job code reusable. Routines are used in various parts of the job design. 95 routines are delivered as part of EPM ETL. These are present in a category called 'EPM_Routines'.
Where can one find the details for all the EPM Routines?	See Appendix C, "ETL Reference Documents," Routine Descriptions, page 671.

ETL Job Process Flow

This table provides answers to questions about the ETL job process flow.

Question	*Answer*
How does EPM manage the process flow for ETL jobs? How are job interdependencies managed?	DataStage Sequence job allows you to specify several jobs to run in controlled manner and can be used to specify different courses of action to take depending on whether a job in the Sequence succeeds or fails. Every ETL load has a Sequence job and each business process within a datamart is provided with a master sequence to trigger all the jobs belonging to it. See Chapter 15, "ETL Configurations," Using the Master Sequencer Utility to Create Master Sequencer Jobs, page 319.
What are the common triggers used in the process flow?	Triggers are used to control the flow of a Sequence job in triggering various other Sequence/Server child jobs. The most commonly used ones are • Failed – Conditional • Warning – Conditional • OK – Conditional • Unconditional

ETL Utilities

This table provides answers to questions about delivered ETL utilities.

Question	*Answer*
What is the Language Swap utility?	If the source database base language is different from the EPM database base language, you must ensure that the EPM base tables have descriptions in EPM base language and the related language table have descriptions in EPM installed foreign language. The Language Swap ETL utility provides this functionality. For more information on the language swap utility, see the chapter *Setting Up Multilanguage Processing and Running the Language Swap Utility* in your warehouse specific PeopleBook (for example, the *PeopleSoft Enterprise Campus Solutions Warehouse PeopleBook*).
What are related language tables and how are they packaged?	In EPM, every table that requires language translation has a corresponding related language table. ETL jobs to populate these language tables are created. These jobs are packaged along with the base table jobs. Running these jobs is optional, since not all of them require the use of multi-language functionality.
What is the Currency Conversion utility?	This utility is used to populate the reporting amount and reporting currency code columns in fact tables in the Multidimensional Warehouse. This population is considered an ETL post process. Before running the ETL, the setup for MDW Currency Conversion definitions should be completed in the PIA pages. For more information on the MDW currency conversion utility, see the chapter *Implementing Currency Conversion for Multiple Currencies* in your warehouse specific PeopleBook (for example, the *PeopleSoft Enterprise Campus Solutions Warehouse PeopleBook*).
What are tree-processing jobs?	If a warehouse or data mart has tree or recursive hierarchy data, the ETL utility to process this data must be triggered. The utility flattens and denormalizes the set of hierarchies. These hierarchy definitions needs to defined in PIA pages before running the ETL jobs. For more information on the MDW tree flattening process, see the chapter *Processing Trees and Recursive Hierarchies* in your warehouse specific PeopleBook (for example, the *PeopleSoft Enterprise Campus Solutions Warehouse PeopleBook*).

Question	*Answer*
What are the Dimension Mapper jobs?	Dimension mapper jobs refer to a set of jobs that enable users to set up Warehouse Business Units and SetIDs. This mapper tool utilizes data from several other tables such as PF_SRC_SETCNTRL, PF_SRC_BU_NAMES, and PF_SRC_BU_ROLES, which are loaded by the dimension mapper jobs. The output tables of the Dimension Mapper tool are PF_SETID_LOOKUP, PF_BUS_UNIT_MAP, BUS_UNIT_TBL_PF, BUS_UNIT_TBL_FS, SETID_TBL, SET_SNTRL_TBL, SET_CNTRL_GROUP, and SET_CNTRL_REC. These tables are used as lookups in ETL job design. See Chapter 14, "Importing Source Business Units into EPM to Create Warehouse Business Units," page 283.

Running ETL Jobs

This table provides answers to questions about running ETL jobs.

Question	*Answer*
How do I identify ETL jobs that are needed based on the license code or my implementation plans?	See Appendix D, "Using the PeopleSoft EPM Lineage Spreadsheets," page 673.
What do the OWS to OWE (D00) jobs do?	The D00 job from OWS to OWE loads data from staging tables to _D00 tables in the OWE database.
What do the OWS to OWE (F00) jobs do?	The F00 job from OWS to OWE loads data from staging tables to _F00 tables in the OWE database.
What do the OWS to MDW (Global Dimensions) jobs do?	The Global Dimension jobs loads data from staging tables to dimension tables. Global Dimensions are dimension tables that are shared across warehouses.
What do the OWS to MDW (Local Dimensions) jobs do?	The Local Dimension jobs loads data from staging tables to dimension tables. Local Dimensions are dimension tables that are shared across different marts in a warehouse.
What do the OWS to MDW (data mart dimensions and facts) jobs do?	The dimension and fact job loads data from staging tables to dimension or fact tables. A dimension contains a key (SID) value and attributes used for slicing and dicing measures located in a fact table.
What do the OWE to MDW (Dimensions and Facts) jobs do?	The dimension and fact job loads data from OWE tables (D00 or F00) to dimension or fact tables. A dimension contains a key (SID) value and attributes used for slicing and dicing measures located in a fact table.

Question	*Answer*
How do uncompiled jobs affect EPM?	All server jobs/sequencers should be compiled before running. Uncompiled jobs will not run and have to be compiled using the Designer prior to running.
In what order must I run different categories of jobs?	See Appendix D, "Using the PeopleSoft EPM Lineage Spreadsheets," page 673.
How are Master Sequences used to load data marts?	• *Loading Sequence for OWS:* To run the OWS jobs, you can use the Master_Run_Utility, which reads the list of OWS jobs from a flat file and executes them in the specific order as mentioned in the input flat file. Using this Master_Run_Utility, you can run Hash Load jobs (first time to create the Hash Files), Base - Sequence Jobs, and Language Sequence jobs. • *Loading Sequence for OWE:* To run the OWE jobs, you can use the Master_Run_Utility, which reads the list of OWS jobs from a flat file and executes them in the specific order as mentioned in the input flat file. Using this Master_Run_Utility, user can run Hash Load jobs (first time to create the Hash Files), D00, F00 and Language D00 jobs. • *Loading Sequence for MDW:* To run the MDW Dimensions and Facts, user can run the Master_Sequence jobs that are present under the Master_Sequence category. Master_Sequence category will be present under all subject areas of the data mart node. Hence, there are four Master Sequence jobs for each subject area: Hash Load Jobs, Base - Dimensions, Facts and Language Dimensions. For Global Dimensions and Local Dimensions you can find 3 Master Sequence; Hash Load Jobs, Base – Dimensions and Language Dimensions. Alternately, you can also use Master_Run_Utility to run the MDW Dimensions and Facts, provided there is a loading sequence flat file present in the DataStage Server.
Can I use the People Tools Process Scheduler to trigger jobs?	No, you cannot use the Process Scheduler to trigger jobs. To schedule jobs, you can use the DataStage Scheduler or any other third party scheduling tool.
When do I use the Master Run utility in the utilities folder?	The Master_Run_Utility can be used to run the set of jobs that are present in a flat file in the DataStage Server. This utility will read the list of jobs that are present in the file and trigger them in a serial mode, by taking care of dependency logic as mentioned in the Input Flat file. Master_Run_Utility can be used to run any jobs in a dependent or independent mode.

Error Handling with ETL Jobs

This table provides answers to questions about error handling with ETL jobs.

Question	*Answer*
What is the Error validation mechanism built into EPM ETL design?	See Chapter 7, "Preparing to Load Source Data Into EPM," Understanding Data Validation and Error Handling in the ETL Process, page 179.
How is Error Validation performed using Error tables?	See Chapter 7, "Preparing to Load Source Data Into EPM," Understanding Data Validation and Error Handling in the ETL Process, page 179.

ETL Job Validation

This table provides answers to questions about ETL job validation.

Question	*Answer*
How do I confirm whether a job has run successfully?	Log onto Datastage director and check the status of the job. If the status of the job is "Finished" then the job competed Successfully with no errors or warnings. If the status is finished then you can go and query your target database and check if the number of rows populated in the target table are correct. If the status of the job is Finished (See Log) then there are some warnings in the job and you need to check the log file to see what the warning is about and fix that particular issue and run it till the job completes with status Finished. Finally if the status is Abort then the job has some Fatal errors. You need to fix those errors and run the job again to make sure the job is in "Finished" status.

Question	Answer
How can I test OWS, OWE, and MDW jobs?	• *OWS:* Run the sequencer and check if the job status of the sequencer is in Finished. Then do a data compare between the source and the target table and make sure the data is matching exactly to the result set you have. • *OWE:* Ensure the OWS data is populated first as the source for OWE jobs is OWS. Run the OWE job and make sure the job is in Finished status. Do a data compare with the source and target database to make sure the data that is populated in the target tables is matching to your expected result set. • *MDW:* Ensure the OWS data is populated first as the source for MDW job is OWS. In some cases, the source can be OWE and in such cases the OWE jobs must be executed first before running the MDW jobs. Run the MDW job and make sure the job is in 'Finished' status. Do a data compare with source and target database and check if the data populated in the target database is matching to your expected results.
How do I validate a job execution by verifying the data?	Perform a Database compare. This can be done by handwritten scripts or by third party DBCompare tools.

Common ETL Issues

This table provides answers to questions about common ETL issues.

Question	Answer
How do you debug a job that aborted or completed with warnings?	See Chapter 10, "Using DataStage Director," Managing Aborted and Failed Jobs, page 252.
What if I drop and recreate an index?	Dropping and recreating an index would increase the ETL performance. However the downside of it is when there are any integrity constraints defined via indexes at the DB level and they not being handled in the ETL application. In such cases, data that gets loaded might not be cleaner and this might produce errors during recreating the index.

Question	***Answer***
How do I resolve the following issue I receive when running any job in DataStage: I receive the error message - *Could not load drsoci.so* when pointed to an Oracle database and the DataStage server is UNIX.	Verify the dsenv file, which is a centralized file for storing environmental variables in the DataStage Server. It resides in $DSHOME, where $DSHOME identifies the DataStage main directory (for example /u1/dsadm/IBM WebSphere/DataStage/DSEngine). The dsenv file is a series of Bourne shell arguments, which are referenced during DataStage server startup and can be referenced by interactive users or other programs or scripts. For a connection using a non-wire protocol driver, you generally need to specify the following in the dsenv file: • Environment variables required by the database client software • Database home location • Database library directory Certain Plug-ins require shared libraries to be loaded and you need to include the library path in an environment variable. The names of the library path environment variables is platform dependent: • Solaris Platform = LD_LIBRARY_PATH • HP-UX Platform = SHLIB_PATH • AIX Platform = LIBPATH • Compaq Tru64 Platform = LD_LIBRARY_PATH • LINUX Platform = LD_LIBRARY_PATH The following provides typical entries for commonly used databases: # Oracle 8i ORACLE_HOME=/space/oracle8i ORAHOME=/space/oracle8i LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib:$ORACLE_HOME/rdbms/lib;export LD_LIBRARY_PATH ORACLE_SID=WSMK5 ORASID=WSMK5 export ORACLE_HOME ORAHOME ORACLE_SID ORASID See *IBM Information Server: Planning Installation and Configuration Guide*

Question	*Answer*
How do I resolve the following issue I receive when running any job in DataStage: I receive the error message - *Could not load drsdb2.so* when pointed to a DB2 database and the DataStage server is UNIX.	Verify the dsenv file, which is a centralised file for storing environmental variables in the DataStage Server. It resides in $DSHOME, where $DSHOME identifies the DataStage main directory (for example /u1/dsadm/IBM WebSphere/DataStage/DSEngine). The dsenv file is a series of Bourne shell arguments, which are referenced during DataStage server startup and can be referenced by interactive users or other programs or scripts. For a connection using a non-wire protocol driver, you generally need to specify the following in the dsenv file: • Environment variables required by the database client software • Database home location • Database library directory Certain Plug-ins require shared libraries to be loaded and you need to include the library path in an environment variable. The names of the library path environment variables is platform dependent: • Solaris Platform = LD_LIBRARY_PATH • HP-UX Platform = SHLIB_PATH • AIX Platform = LIBPATH • Compaq Tru64 Platform = LD_LIBRARY_PATH • LINUX Platform = LD_LIBRARY_PATH The following provides typical entries for commonly used databases: #DB2 6.1 DB2DIR=/opt/IBMDB2/V6.1;export DB2DIR DB2INSTANCE=DB2inst1; export DB2INSTANCE INSTHOME=/export/home/DB2inst1;export INSTHOME PATH=$PATH:$INSTHOME/sqllib/bin:$INSTHOME/sqllib/adm:$INSTHOME/sqllib/misc export PATH LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$INSTHOME/sqllib/lib;export LD_LIBRARY_PATH THREADS_FLAG=native;export THREADS_FLAG

Question	*Answer*
How do I switch to a new project when the warehouse tables have already been loaded with data?	There might be a need to switch to new project when the warehouse tables have already been loaded for some time. In such cases, there is some project specific control data that must be restored onto the new project. For this purpose, it is always a good idea to backup this control data at some regular intervals of time after significant chunk of ETL loading gets completed. PeopleSoft delivers utilities that the backup/recovery process. Refer the section 'Running Datastage Project Utilities' and the following subsections in the PeopleSoft EPM Red Paper: ETL Implementation Steps (found in My Oracle Support) procedure to run these utilities: • Backup_SurrogateKey_HashFile – Utility • Backup_DateTime_HashFiles – Utility • Recovery_DateTime_HashFiles – Utility • Recovery_SurrogateKey_HashFile – Utility
If a job aborts after half of the one million rows are written to the tables, what will happen?	If the Transaction size is selected as zero and if the job aborts in the middle, then the job will rollback the transactions since it follows the principle of Two-way commit. If the Transaction size is anything other than zero and if the job fails in the middle, then the job will perform commits for the number of rows that processed till the error message.
How do I report an issue with a job log for a job that completed with warnings or errors?	Report an issue to Global Customer Support. To report an issue, you must include the job log of the last run. In DataStage Designer, view the log for the job in detailed view mode. Select Project, Print from the menu. In the Print dialog box, select the *All entries*, *Full details*, and *Print to file* options. Click OK, name the file, and send the log along with your issue description and other pertinent information.

Configuring Delivered ETL Content

This table provides answers to questions about configuring and enhancing delivered ETL content.

Question	*Answer*
What are some of the configurations I may want to perform?	• Convert an incremental load job to a destructive load job. • Parameterize the hash file directory for user defined directories that store hashed files (rather than the default project directory). • Remove lookups that do not pertain to your business. • Configure jobs for initial load.
How do I convert an incremental load job to a destructive load job?	See Chapter 15, "ETL Configurations," Converting Incremental Load Jobs to Destructive Load Jobs, page 326.
How do I run ETL jobs in destructive mode after some have already run as incremental loads?	See Chapter 15, "ETL Configurations," Converting Incremental Load Jobs to Destructive Load Jobs, page 326.
How do I add a new environmental variable?	See Chapter 15, "ETL Configurations," Adding New Environmental Variables, page 332.
What if I make a change that requires a new attribute in a dimension table?	If there is change in the data model with respect to a new addition of attribute to the EPM database, then you has to update the corresponding dimension job to incorporate this new attribute, otherwise the job will fail. If there is no source for this new attribute, then in the dimension job you can have a Default value assigned to this new attribute by using a routine that is delivered.
What if I make a change that requires a new dimension in a fact table?	If a new dimension key is added to a fact table in the database, then this is a change to the data model. Since the database has an additional dimension key for the fact table, this will result in changes to the ETL job. If this is a new dimension, then a new job has to be developed for this new dimension. Fact job must be updated accordingly with the correct dimension key and Corresponding SID population in the Fact table.
What if I make a change that requires a new measure in a fact table?	If a new measure is added to a fact table in the database, then this is a change to the data model. Since the database has an additional measure for the fact table, this will result in changes to the ETL job. Fact job must be updated accordingly with the correct measure getting assigned to the value that is either coming from the source directly or applying any logic that is required for this measure to be populated as per your requirements.
What if I make a change that requires a new dimension table?	A new ETL job has to be developed for this new dimension table as per the requirements.
What if I make a change that requires a new fact table?	A new ETL job has to be developed for this new fact table as per the requirements.

Question	***Answer***
What are some techniques I can use to enhance ETL performance?	• *InterProcess Stage:* The IPC stage is used to implement pipeline parallelism. The IPC stage can be used to explicitly de-link two passive stage activities to run in separate processes. As a good practice an IPC stage can be inserted before a database write stage. • *Link Partitioner/Link Collector Stages:* The link partitioner and link collector can be used in conjunction to implement partition parallelism. Usage of link partitioner and link collector can be envisaged where specific requirement exists to further enhance job performance. This can be used most effectively to run in parallel processes that take a lot of time. The functionality that is to be run in parallel is typically identical and can be made into a local container. This way any changes to the common parallel part of the job need to be effected only at a single place.
How can I enhance the parallel processing capabilities of ETL server jobs?	• *InterProcess Stage:* The IPC stage is used to implement pipeline parallelism. The IPC stage can be used to explicitly de-link two passive stage activities to run in separate processes. As a good practice an IPC stage can be inserted before a database write stage. • *Link Partitioner/Link Collector Stages:* The link partitioner and link collector can be used in conjunction to implement partition parallelism. Usage of link partitioner and link collector can be envisaged where specific requirement exists to further enhance job performance. This can be used most effectively to run in parallel processes that take a lot of time. The functionality that is to be run in parallel is typically identical and can be made into a local container. This way any changes to the common parallel part of the job need to be effected only at a single place. See *WebSphere DataStage Development: Designer Client Guide*

Appendix C

ETL Reference Documents

This appendix provides the following ETL reference documents:

- DSX files import description.
- Environmental parameters information.
- Parameter and source data files information.
- Routine descriptions and information.

General DSX Files					
DSX Import Order	**General / Warehouse Specific**	**Type**	**DSX File Name**	**Location**	**DSX file Description**
1	General	Common Utilities	Common_Utilities.dsx	<PSHOME>\SRC\ETL	This dsx file will have all the routines, shared containers, shared lookups, reusable jobs and utilities (including the jobs related to Language swap and Tree processing) Will also include Common Dimension jobs.
2	General	Common Jobs for SETUP_OWE, SETUP_DIMENSION_MAPPER, COMMON_DIMENSIONS, GLOBAL_DIMENSIONS	Common.dsx	<PSHOME>\SRC\ETL	All jobs relating to SETUP- DIMENSION_MAPPER (both Base and Related Language, if applicable), All jobs relating to SETUP-OWE (both Base and Related Language), sourced from the OWS tables. All jobs relating to MDW COMMON DIMENSIONS (both Base and Related Language), sourced from OWS tables that get data from Enterprise transaction systems. All jobs relating to MDW COMMON DIMENSIONS (both Base and Related Language), sourced from OWE tables. All jobs relating to MDW GLOBAL_DIMENSIONS (both Base and Related Language), sourced from OWS tables that get data from Enterprise transaction systems. These dimensions are shared across various warehouse(s) and are however owned by different ware houses. For example D_PERSON is a Shared dimension owned by the HCM warehouse, but is being used by other warehouses, both as a lookup and also gets data from OWS tables (that get data from other source transation systems)
3	General	All E OWE Jobs for all Warehouses	OWE.dsx	<PSHOME>\SRC\ETL	All E-OWE Jobs, (there will be no categories for functional areas and no categories for D00's /F00's). (both Base and Related Language for D00's)

CRM Warehouse - DSX Files description

DSX Import Order	Product Code/SKU	Type	DSX File Names	Location	DSX File Description
1	CRM - SETUP & OWS	SETUPS and OWS Jobs	WCR_OWS.dsx	<PSHOME>\SRC\ETL	This will have the jobs relating to E - setups and E - general tables in the OWS (both Base and Language) for the CRM Warehouse
2	CRM - Local Dimensions	LOCAL DIMENSIONS	WCR_MDW_LOCAL_DIMS.dsx	<PSHOME>\SRC\ETL	All jobs relating to Local dimensions (Base and Language) for CRM warehouse in E
3	CTM - Customer Mart	MDW Jobs (SKU)	WCR_CUSTOMER_MART.dsx	<PSHOME>CTM\BSE\SRC\ETL	All jobs relating to Dimensions (Base and Language) for this particular SKU and Facts. This file will also contain the currency conversion jobs for the facts related to this SKU. There is no specific order in how you implement any of the SKU's related to WCR Marts.
4	MM - Marketing Mart	MDW Jobs (SKU)	WCR_MARKETING_MART.dsx	<PSHOME>MM\BSE\SRC\ETL	As above
5	SLM - Sales Mart	MDW Jobs (SKU)	WCR_SALES_MART.dsx	<PSHOME>SLM\BSE\SRC\ETL	As above
6	SVM - Services Mart	MDW Jobs (SKU)	WCR_SERVICE_MART.dsx	<PSHOME>SVM\BSE\SRC\ETL	As above

Campus Solutions Warehouse - DSX Files description					
DSX Import Order	**Product Code/SKU**	**Type**	**DSX File Names**	**Location**	**DSX File Description**
1	CS - Setups & OWS	Setup and OWS Jobs	WCS_OWS.dsx	<PSHOME>\SRC\ETL	This contains the jobs related to setup and general tables in the OWS (Base) for the CS Warehouse
2	CS - Local Dimensions	LOCAL DIMENSIONS	WCS_MDW_LOCAL_DIMS.dsx	<PSHOME>\SRC\ETL	All jobs related to Local dimensions (Base) for the CS warehouse.
3	SFM - Student Financials Mart	MDW Jobs (SKU)	WCS_STUDENT_FINANCIALS_MART.dsx	<PSHOME>\SFM\BSE\SRC\ETL	All jobs related to the Dimensions (Base) for this particular SKU and Facts. This file will also contain the currency conversion jobs for the facts related to this SKU. There is no specific order in how you implement any of the SKU's related to CS Marts.
4	STM - Student Records Mart	MDW Jobs (SKU)	WCS_STUDENT_RECORDS_MART.dsx	<PSHOME>\STM\BSE\SRC\ETL	As above
5	ADM - Admissions and Recruting Mart	MDW Jobs (SKU)	WCS_ADMISSIONS_AND_RECRUITING_MART.dsx	<PSHOME>\ADM\BSE\SRC\ETL	As above
6	CCM - Campus Community Mart	MDW Jobs (SKU)	WCS_CAMPUS_COMMUNITY_MART.dsx	<PSHOME>\ADM\BSE\SRC\ETL	As above
7	CS - OWS Audit jobs	OWS Audit Jobs	WCS_OWS_AUDIT.dsx	<PSHOME>\SRC\ETL	This is optional. This DSX contains audit jobs for the CS Warehouse. These audit jobs are an alternative to the staging jobs which use CRC incremental logic. These audit jobs have been implemented using audit records which are built on the source system. They perform much faster than the staging jobs which use CRC logic.

Financials Warehouse - DSX Files description

DSX Import Order	Product Code/SKU	Type	DSX File Names	Location	DSX File Description
1	EFM - OWS	OWS Jobs	WFN_OWS.dsx	<PSHOME>\SRC\ETL	This contains the jobs related to general tables in the OWS (both Base and Language) for the EFM Warehouse
2	EFM and SCM Setups	Setups	WFN_WSC_OWS_SETUP.dsx	<PSHOME>\SRC\ETL	This dsx file contains the jobs relating to setup for FMS and SCM (both Base and Language). **Note: If you already imported SCM warehouse, you might have imported this dsx. In that case, you can ignore this file.**
3	EFM - Local Dimensions	LOCAL DIMENSIONS	WFN_MDW_LOCAL_DIMS.dsx	<PSHOME>\SRC\ETL	All jobs related to Local dimensions (Base and Language) for EFM warehouse
4	FMA - General Ledger and Profitability Mart	MDW Jobs (SKU)	WFN_GENERAL_LEDGER_AND_PROFITABILITY_MART.dsx	<PSHOME>\FMA\BSE\SRC\ETL	All jobs related to Dimensions (Base) for this particular SKU and Facts. This file will also contain the currency conversion jobs for the facts related to this SKU. There is no specific order in how you implement any of the SKU's related to FMS Marts.
5	PYM - Payables Mart	MDW Jobs (SKU)	WFN_PAYABLES_MART.dsx	<PSHOME>\PYM\BSE\SRC\ETL	As above
6	RBM - Receivables Mart	MDW Jobs (SKU)	WFN_RECEIVABLES_MART.dsx	<PSHOME>\RBM\BSE\SRC\ETL	As above
7	WES - ESA Mart	MDW Jobs (SKU)	WFN_ESA_MART.dsx	<PSHOME>\WES\BSE\SRC\ETL	As above
8	EFM - OWS Audit jobs	OWS Audit Jobs	WFN_OWS_AUDIT.dsx	<PSHOME>\SRC\ETL	This is optional. This DSX contains audit jobs for the FMS Warehouse. These audit jobs are an alternative to the staging jobs which use CRC incremental logic. These audit jobs have been implemented using audit records which are built on the source system. They perform much faster than the staging jobs which use CRC logic.

HCM Warehouse - DSX Files description

DSX Import Order	Product Code/SKU	Type	DSX File Names	Location	DSX File Description
1	HCM - Setups & OWS	SETUP and OWS Jobs	WHR_OWS.dsx	<PSHOME>\SRC\ETL	This contains the jobs related to setup and general tables in the OWS (both Base and Language) for the HCM Warehouse & Applications.
2	HCM - Local Dimensions	LOCAL DIMENSIONS	WHR_MDW_LOCAL_DIMS.dsx	<PSHOME>\SRC\ETL	All jobs related to Local dimensions (Base and Language) for HCM warehouse
3	CPM - Compensation Mart	MDW Jobs (SKU)	WHR_COMPENSATION_MART.dsx	<PSHOME>CPM\BSE\SRC\ETL	All jobs related to Dimensions (Base and Language) for this particular SKU and Facts. This file will also contain the currency conversion jobs for the facts related to this SKU. There is no specific order in how you implement any of the SKU's related to HCM Marts.
4	LDM - Learning and Development Mart	MDW Jobs (SKU)	WHR_LEARNING_AND_DEVELOPMENT_MART.dsx	<PSHOME>LDM\BSE\SRC\ETL	As above
5	RTM - Recruting Mart	MDW Jobs (SKU)	WHR_RECRUITING_MART.dsx	<PSHOME>RTM\BSE\SRC\ETL	As above
6	WPM - Workforce Profile Mart	MDW Jobs (SKU)	WHR_WORKFORCE_PROFILE_MART.dsx	<PSHOME>WPM\BSE\SRC\ETL	As above

SCM Warehouse - DSX Files description

DSX Import Order	Product Code/SKU	Type	DSX File Names	Location	DSX File Description
1	SCM - OWS	SETUP and OWS Jobs	WSC_OWS_E.dsx	<PSHOME>\SRC\ETL	This contains the jobs related to general tables in the OWS (both Base and Language) for the SCM Warehouse
2	EFM and SCM Setups	Setups	WFN_WSC_OWS_SETUP.dsx	<PSHOME>\SRC\ETL	This dsx file contains the jobs relating to setup for FMS and SCM (both Base and Language). **Note: If you already imported FMS warehouse, you might have imported this dsx. In that case, you can ignore this file.**
3	SCM - Local Dimensions	LOCAL DIMENSIONS	WSC_MDW_LOCAL_DIMS.dsx	<PSHOME>\SRC\ETL	All jobs relating to Local dimensions (Base and Language) for SCM warehouse
4	IA - Inventory Mart	MDW Jobs (SKU)	WSC_INVENTORY_MART.dsx	<PSHOME>\IA\BSE\SRC\ETL	All jobs relating to Dimensions (Base and Language) for this particular SKU and Facts. This file will also contain the currency conversion jobs for the facts related to this SKU. There is no specific order in how you implement any of the SKU's related to SCM Marts.
5	MA - Manufacturing Mart	MDW Jobs (SKU)	WSC_MANUFACTURING_MART.dsx	<PSHOME>\MA\BSE\SRC\ETL	As above
6	PSMA - Procurement Mart	MDW Jobs (SKU)	WSC_PROCUREMENT_MART.dsx	<PSHOME>\PSMA\BSE\SRC\ETL	As above
7	SMA - Fulfillment and Billing Mart	MDW Jobs (SKU)	WSC_FULFILLMENT_AND_BILLING_MART.dsx	<PSHOME>\SMA\BSE\SRC\ETL	As above
8	SPM - Supply Chain Planning Mart	MDW Jobs (SKU)	WSC_SUPPLY_CHAIN_PLANNING_MART.dsx	<PSHOME>\SPM\BSE\SRC\ETL	As above
9	SRM - Spend Mart	MDW Jobs (SKU)	WSC_SPEND_MART.dsx	<PSHOME>\SRM\BSE\SRC\ETL	As above

Required Parameters for All EPM Warehouses					
Parameter Name	**Type**	**Prompt**	**Description**	**Example for Default Values**	**Notes**
DATA_ORIGIN	String	Data Origin	For triggering the jobs moving the data from source(OWS) should have DATA_ORIGIN = 'S'. For triggering the jobs moving the data from OWE(EPM) should use 'E'	S	To be used in all server and sequencer jobs
ERR_VALIDATE	String	Error Validation (Y/N)	Vaue 'Y' is for validating the lookups. 'N' for no validation	Y	use this in every dimension/fact server/sequence jobs
ERR_THRESHOLD	String	Error Threshold	The row limit failing the lookups after which job aborts.	1000000	use this in every dimension/fact server/sequence jobs
PARAM_FILE_DIR	String	Enter Parameter File Directory	The directory or path in which all Parameter Files are kept.	C:\Ascential\Datastage\TestProj\	usually used in E1 jobs which has UDC values to be defined
SOURCE_FILE_DIR	String	Source File Directory	The directory or path in which all Source Data Files(Flat Files)are kept.	C:\Ascential\Datastage\TestProj\	Used for source data files if any.
HASHED_FILE_DIRECTORY	String	Hash file directory	The directory where all the Hashed Files are created and stored	C:\Hash_Dir	
SURVEY_SRC_SYS_ID	String	Survey Source System Identifier	This is the Survey Source System Identifier	C:\\Ascential\DataStage\Projects\EPM9_DVL\ENV_FILES\	
BACKUP_FILE_DIR	String	Backup Hash File Directory	The directory or path for the backup of Hashed files.	D:\\Ascential\DataStage\Projects\EPM9_DVL\	
OWS_DBTYPE	String	OWS DB Type	OWS - Database Type. Several values are Oracle, IBM DB2, MSSQL Server	MSSQL Server	
OWS_DBCONNECTION	String	OWS DB Connection	ODBC Connection Name for OWS database	ET890DVL	
OWS_SCHEMA	String	OWS Schema ID	Schema name for OWS Database	dbo.	
OWS_USERNAME	String	OWS Username	Username for OWS Database	sadvl	
OWS_PASSWORD	Encrypted	OWS Password	Password for OWS Database	h0tel	
OWS_AS	String	OWS Array Size	Array size on OWS side connection	32767	
OWS_TZ	String	OWS Transaction Size	Transaction size on OWS side connection	0	
OWS_IPC_BUF_SIZE	String	OWS IPC Buffer Size	IPC Buffer Size on OWS side connection	512	
OWS_IPC_TIMEOUT	String	OWS IPC Time Out	IPC timeout on OWS side connection	500	
OWE_DBTYPE	String	OWE DB Type	OWE - Database Type. Several values are Oracle, IBM DB2, MSSQL Server	MSSQL Server	
OWE_DBCONNECTION	String	OWE DB Connection	ODBC Connection Name for OWE database	ET890DVL	
OWE_SCHEMA	String	OWE Schema ID	Schema name for OWE Database	dbo.	
OWE_USERNAME	String	OWE Username	Username for OWE Database	sadvl	
OWE_PASSWORD	Encrypted	OWE Password	Password for OWE Database	h0tel	
OWE_AS	String	OWE Array Size	Array size on OWE side connection	32767	
OWE_TZ	String	OWE Transaction Size	Transaction size on OWE side connection	0	
OWE_IPC_BUF_SIZE	String	OWE IPC Buffer Size	IPC Buffer Size on OWE side connection	512	
OWE_IPC_TIMEOUT	String	OWE IPC Time Out	IPC timeout on OWE side connection	500	

MDW_DBTYPE	String	MDW DB Type	MDW - Database Type. Several values are Oracle, IBM DB2, MSSQL Server	MSSQL Server	
MDW_DBCONNECTION	String	MDW DB Connection	ODBC Connection Name for MDW database	ET890DVL	
MDW_SCHEMA	String	MDW Schema ID	Schema name for MDW Database	dbo.	
MDW_USERNAME	String	MDW Username	Username for MDW Database	sadvl	
MDW_PASSWORD	Encrypted	MDW Password	Password for MDW Database	h0tel	
MDW_AS	String	MDW Array Size	Array size on MDW side connection	32767	
MDW_TZ	String	MDW Transaction Size	Transaction size on MDW side connection	0	
MDW_IPC_BUF_SIZE	String	MDW IPC Buffer Size	IPC Buffer Size on MDW side connection	512	
MDW_IPC_TIMEOUT	String	MDW IPC Time Out	IPC timeout on MDW side connection	500	
SID_UNIQUENESS	String	Surrogate Key Uniqueness	Specifies the surrogate uniqueness for a Warehouse. For uniquess across warehouse use 'W'. For uniqueness across dimension, use 'D'	D	All dimension server/sequence jobs has it
SCD_TYPE	String	Slowly changing dimension type	Value to specify the slowly changing dimension type.(1 or 2)		Value to determine the SCD type(Note: Type 1 is implemented out of the box)

Warehouse and Source System Specific Parameters					
Parameter Name	**Type**	**Prompt**	**Description**	**Example for Default Values**	**Comments**
CRM_SRC_DBTYPE	String	CRM Source DB Type	CRM - Database Type. Several values are Oracle, IBM DB2, MSSQL Server	MSSQL Server	
CRM_SRC_DBCONNECTION	String	CRM Source DB Connection	ODBC Connection Name for CRM database	CR890P23	
CRM_SRC_SCHEMA	String	CRM Source Schema ID	Schema name for CRM Database	dbo.	
CRM_SRC_USERNAME	String	CRM Source Username	Username for CRM Database	satst	
CRM_SRC_PASSWORD	Encrypted	CRM Source Password	Password for CRM Database	satst	
CRM_AS	String	CRM Source Array Size	Array size on CRM side connection	32767	
CRM_SRC_SYS_ID	String	CRM Source System Identifier	The Source System ID of CRM database	CRM, CRUSA, etc.	All staging jobs for CRM have it
CRM_LOADTYPE	String	CRM Datamart Load Type	Load Type for the CRM DataMart		
CRM_LOG_DIR	String	CRM Datamart Log File Directory	The directory or path for the Log files	C:\Ascential\Datastage\LogFiles\	
CRM_SRC_TIMEZONE	String	CRM Source Timezone	Timezone of CRM database	PST, IST etc..	
CRM_IPC_BUF_SIZE	String	CRM IPC Buffer Size	IPC Buffer Size on CRM side connection	512	
CRM_IPC_TIMEOUT	String	CRM IPC Time out	IPC timeout on CRM side connection	500	
HCM_SRC_DBTYPE	String	HCM Source DB Type	HCM - Database Type. Several values are Oracle, IBM DB2, MSSQL Server	MSSQL Server	
HCM_SRC_DBCONNECTION	String	HCM Source DB Connection	ODBC Connection Name for HCM database	HC890P21	
HCM_SRC_SCHEMA	String	HCM Source Schema ID	Schema name for HCM Database	dbo.	
HCM_SRC_USERNAME	String	HCM Source Username	Username for HCM Database	satst	
HCM_SRC_PASSWORD	Encrypted	HCM Source Password	Password for HCM Database	satst	
HCM_AS	String	HCM Source Array Size	Array size on HCM side connection	32767	
HCM_SRC_SYS_ID	String	HCM Source System Identifier	The Source System ID of HCM database	HCM, HCUSA etc.	
HCM_LOADTYPE	String	HCM Datamart Load Type	Load Type for the HCM DataMart		
HCM_LOG_DIR	String	HCM Datamart Log File Directory	The directory or path for the Log files	C:\Ascential\Datastage\LogFiles\	
HCM_SRC_TIMEZONE	String	HCM Source Timezone	Timezone of HCM database	PST, IST etc..	
HCM_IPC_BUF_SIZE	String	HCM IPC Buffer Size	IPC Buffer Size on HCM side connection	512	
HCM_IPC_TIMEOUT	String	HCM IPC Time out	IPC timeout on HCM side connection	500	
HCM_TZ	String	HCM Source Transaction Size	Transaction size for the HCM source	0	
ELM_SRC_DBTYPE	String	ELM Source DB Type	ELM - Database Type. Several values are Oracle, IBM DB2, MSSQL Server	MSSQL Server	
ELM_SRC_DBCONNECTION	String	ELM Source DB Connection	ODBC Connection Name for ELM database	HC890P21	
ELM_SRC_SCHEMA	String	ELM Source Schema ID	Schema name for ELM Database	dbo.	
ELM_SRC_USERNAME	String	ELM Source Username	Username for ELM Database	satst	
ELM_SRC_PASSWORD	Encrypted	ELM Source Password	Password for ELM Database	satst	
ELM_AS	String	ELM Source Array Size	Array size on ELM side connection	32767	
ELM_SRC_SYS_ID	String	ELM Source System Identifier	The Source System ID of ELM database	ELUSA, ELM, etc.	
ELM_LOADTYPE	String	ELM Datamart Load Type	Load Type for the ELM DataMart		

Parameter Name	Type	Prompt	Description	Example for Default Values	Comments
ELM_LOG_DIR	String	ELM Datamart Log File Directory	The directory or path for the Log files	C:\Ascential\Datastage\LogFiles\	
ELM_SRC_TIMEZONE	String	ELM Source Timezone	Timezone of ELM database	PST, IST etc..	
ELM_IPC_BUF_SIZE	String	ELM IPC Buffer Size	IPC Buffer Size on ELM side connection	512	
ELM_IPC_TIMEOUT	String	ELM IPC Time out	IPC timeout on ELM side connection	500	
FSCM_SRC_DBTYPE	String	FSCM Source DB Type	FSCM - Database Type. Several values are Oracle, IBM DB2, MSSQL Server	MSSQL Server	
FSCM_SRC_DBCONNECTION	String	FSCM Source DB Connection	ODBC Connection Name for FSCM database	E890P21	
FSCM_SRC_SCHEMA	String	FSCM Source Schema ID	Schema name for FSCM Database	dbo.	
FSCM_SRC_USERNAME	String	FSCM Source Username	Username for FSCM Database	satst	
FSCM_SRC_PASSWORD	Encrypted	FSCM Source Password	Password for FSCM Database	satst	
FSCM_AS	String	FSCM Source Array Size	Array size on FSCM side connection	32767	
FSCM_SRC_SYS_ID	String	FSCM Source System Identifier	The Source System ID of FSCM database	FSUSA, FSCM, etc.	
FSCM_LOADTYPE	String	FSCM Datamart Load Type	Load Type for the FSCM DataMart		
FSCM_LOG_DIR	String	FSCM Datamart Log File Directory	The directory or path for the Log files	C:\Ascential\Datastage\LogFiles\	
FSCM_SRC_TIMEZONE	String	FSCM Source Timezone	Timezone of FSCM database	PST, IST etc..	
FSCM_IPC_BUF_SIZE	String	FSCM IPC Buffer Size	IPC Buffer Size on FSCM side connection	512	
FSCM_IPC_TIMEOUT	String	FSCM IPC Time out	IPC timeout on FSCM side connection	500	
FSCM_TZ	String	FSCM Source Transaction Size	Transaction size for the FSCM source	0	
FSCM_SRC_XML_DIR	String	FSCM Source XML Directory	The directory or path for the XML files	C:\SRCXMLfiles\	
CS_SRC_DBTYPE	String	CS Source DB Type	CS - Database Type. Several values are Oracle, IBM DB2, MSSQL Server	MSSQL Server	
CS_SRC_DBCONNECTION	String	CS Source DB Connection	ODBC Connection Name for CS database	CS890P21	
CS_SRC_SCHEMA	String	CS Source Schema ID	Schema name for CS Database	dbo.	
CS_SRC_USERNAME	String	CS Source Username	Username for CS Database	satst	
CS_SRC_PASSWORD	Encrypted	CS Source Password	Password for CS Database	satst	
CS_AS	String	CS Source Array Size	Array size on CS side connection	32767	
CS_SRC_SYS_ID	String	CS Source System Identifier	The Source System ID of CS database	CSUSA, CS etc.	
CS_LOAD_TYPE	String	CS Datamart Load Type	Load Type for the CS DataMart		
CS_LOG_DIR	String	CS Datamart Log File Directory	The directory or path for the Log files	C:\Ascential\Datastage\LogFiles\	
CS_IPC_BUF_SIZE	String	CS IPC Buffer Size	IPC Buffer Size on CS side connection	512	
CS_IPC_TIMEOUT	String	CS IPC Time out	IPC timeout on CS side connection	500	
CS_FROM_DATE	String	Enter begin date for snapshot	Specifies the start date for snapshot creation	1753-01-01 00:00:00	

Parameter Name	Type	Prompt	Description	Example for Default Values	Comments
CS_TO_DATE	String	Enter end date for snapshot	Specifies the end date for snapshot creation	9999-31-12 00:00:00	
CS_TRANZ_DT	String	CS Transaction Date	Is used to assign the date to the snapshot. This is converted to Date SID in the target fact	1753-01-01 00:00:00	
CS_ACAD_CAR_LST	String	Enter the Academic Career List	Lists the Academic Careers for which the Institution Summary Fact will hold information.	'GRAD','PGRD','RHCH','TECH','UGRD'	
CS_ACN_RSN_CODES	String	Enter the Actions and Action Reason Code(concatenated)	Lists the Action Code and Action Reason Code concatenated as a single word. For Eg. Action Code is 'DISC' and the Action reason Code is 'PDIS'. The value for CS_ACN_RSN_CODES will be 'DISCPDIS'.	'DISCPDIS','DATAPDIS','DISCARMF','LEAVARMF','DISCFASR','LEAVFASR','DISOFCM','LEAVOFCM','DISCDEAT'	
CS_FT_IND	String	CS FT IND	Lists the different values which consider a particular Academic Load as a Full time Academic load.	'F'	
CS_PT_IND	String	CS PT IND	Lists the different values which indicate if the Student has opted for a Part time Academic load.	'P','H','L'	
CS_TZ	String	CS Transaction Size	Transaction size on Source side connection	0	

General Parameters (Staging jobs, dims/facts, D00s/F00s)			
Parameter Name	**Type**	**Prompt**	**Comments**
LastModifiedDateTime	String	Last Modified Date Time	This will be used in every DTTM incremental E-Staging/Dimension and E1-dimension server job
BATCH_SID	Integer	Batch ID	This is required in each server and sequencer job. Default value should necessarily be 0 for sequence jobs.
MinDate	String	Minimum Date	Required in every job whose target has the DATE_RANGE_SBR
MaxDate	String	Maximum Date	Required in every server job whose target has the DATE_RANGE_SBR
Note: There are several other job specific parameters whose values are obtained from the routine call or parameter file. They are supplied to server job from the sequences.			

File Type	File Location	EPM Warehouse	Used for the jobs in the dsx files below	Notes
Source Data Files				
SCP3.0_Base_Model_XML_FILES	\SRC\ETL	SCM	WSC_OWS	Used as sources for the jobs that use XML as Sources in Supply Chain OWS Jobs.
SURVEY_Flat_Files	\SRC\ETL	HCM	OWE	Source file for Survey OWE jobs in HCM warehouse
Parameter Files				
FileParam_FMS	\SRC\ETL	FMS	WFN_GENERAL_LEDGER_AND_PROFITABILITY_MART	
FileParam_HCM	\SRC\ETL	HCM	OWE	Used for E OWE jobs in HCM warehouse
CS_ARCHIVED_LIST_SETUP	\SRC\ETL	CS	WCS_OWS	Used for Delete Startegy
CS_HANDLEDELETES_SETUP	\SRC\ETL	CS	WCS_OWS	Used for Handling Source Archives

Routine Name	Category	Short Description
GetSourceRowCount	EPM_Routines\DeleteStrategy	Gets the Source Row Count for the given Job.
RtnBuildSourceQuery	EPM_Routines\DeleteStrategy	Build Source Query Dynamically.
RtnConvertEPMStd	EPM_Routines\DeleteStrategy	Convert the Source Value to EPM Standards
RtnDeleteHashedRecords	EPM_Routines\DeleteStrategy	Delete a Record from the given Hashed File
RtnGetDelFlag	EPM_Routines\DeleteStrategy	This Routine is used to retrive the Delete Flag
RtnGetJobStartDTTM	EPM_Routines\DeleteStrategy	This Routine is used to retrive the Job Start Datetime
RtnStoreJobStartDTTM	EPM_Routines\DeleteStrategy	Stores the Job Start DateTimeStamp to the Hashed File - HASH_JOBSTARTDTTM
SBRtnWriteDelFlag	EPM_Routines\DeleteStrategy	Subroutine to Store the Delete Flag
AddToDate	EPM_Routines\Generic	Adds the specified amount to the date. The field to which this amount is to be added is specified by the format argument.
ClearHashFile	EPM_Routines\Generic	Clear the Hash File
ClearJobLog	EPM_Routines\Generic	Clear Job Log:- This utility will clear the log file of a job
DateDiff	EPM_Routines\Generic	Gives the number of days between two dates
DateDiffinMin	EPM_Routines\Generic	Gives the total time in minutes between the two dates
DateNotNullable	EPM_Routines\Generic	Returns Not Nullable Dates
DateToDateSIDDefault	EPM_Routines\Generic	Convert Date To Date Sid
ForceAbort	EPM_Routines\Generic	Logs a fatal error message in a job's log file and aborts the job.
ForceWarn	EPM_Routines\Generic	Logs a warning message in a job's log file.
GetAdmFunnel	EPM_Routines\Generic	This routine is used to Generate an SQL Where clause for J_Fact_PS_X_ADM_FUNNEL1 and J_Fact_PS_X_ADM_FUNNEL2 server jobs based on supplied Institution,Academic Career and Adit Term .
GetBURole	EPM_Routines\Generic	Fetches the BU_ROLE corresponding to a specific table name
GetCharDefault	EPM_Routines\Generic	Gets Character type default value for Jobs.
GetCurrentDate	EPM_Routines\Generic	Get the current date
GetCurrentDateTime	EPM_Routines\Generic	Returns the current Date and Time.
GetDateDefault	EPM_Routines\Generic	Gets date type default value for Jobs.
GetDateSIDDefault	EPM_Routines\Generic	Returns Default DATE SID Value
GetEwSrcSysId	EPM_Routines\Generic	Gets the SRC_SYS_ID from HASH_PS_EW_OPTIONS hash file
GetFactMaxRecordSID	EPM_Routines\Generic	Gets the Last update time for the given Job.Used for Integer datatype.
GetFinalDateMonth	EPM_Routines\Generic	Gets the last Date of the month (only the Date Part) of the given input date column.
GetFinalDateofPreviousMonth	EPM_Routines\Generic	Gets the Last Date of the Previous Month for a given Job (unless it itself is the last date)
GetFirstDateMonth	EPM_Routines\Generic	Gets the First Date of the Month of the given input date column.
GetJobReport	EPM_Routines\Generic	This routine will generate the Job report for a given job
GetLastRecordID	EPM_Routines\Generic	The routine is used to retrieve the MaxRecordID stored in a UniVerse file.

Routine Name	Category	Short Description
GetLastUpdDateTime	EPM_Routines\Generic	Gets the Last update time for the given Job.Used for Timestamp datatype
GetMaxDate	EPM_Routines\Generic	Get the Max Date.
GetMaxValueDecimal	EPM_Routines\Generic	Gives the Variable value stored in a record named after the supplied argument, to use from a UniVerse file
GetMinDate	EPM_Routines\Generic	Gets the Minimum Date.
GetNAID	EPM_Routines\Generic	Gets the not available ID. Used for Dimension lookup - Not Available Row
GetNARowLkpKeys	EPM_Routines\Generic	Gets the lookup key values - Used for Dimension lookup - Not Available Row
GetNextBatchNumber	EPM_Routines\Generic	Used for BATCH_SID generation.
GetNextBatchNumberParallel	EPM_Routines\Generic	Used for BATCH_SID generation.
GetNumDefault	EPM_Routines\Generic	Gets Number type default value for Jobs.
GetOWECharDefault	EPM_Routines\Generic	Gets Character type default value for OWE Jobs.
GetOWEDateDefault	EPM_Routines\Generic	Gets Date type default value for OWE Jobs.
GetSIDDefault	EPM_Routines\Generic	Gets SID default value for Jobs.
GetSnapDate	EPM_Routines\Generic	Gets the Last Date of the Previous Month for a given Job (unless it itself is the last date)
GetSnapDtSID	EPM_Routines\Generic	Gets the DAY_SID
KeyMgtGetNextValueConcurrentBATCH	EPM_Routines\Generic	Used for BATCH_SID Generation
LedgerAvgCal	EPM_Routines\Generic	This routine is used to compute Rolling Averages for loading F_LEDGER for E
ManageHashFiles	EPM_Routines\Generic	This routine can either clear the Hash file content or delete the Hash file itself based on the second parameter, ManageType
NullToCharDefault	EPM_Routines\Generic	Replace NULL characters to Default Value(Dash)
NullToDateDefault	EPM_Routines\Generic	Replace NULL to Default Value(1753-01-01 00:00:00)
NullToNumDefault	EPM_Routines\Generic	Replace NULL to Default Value(zero)
NullToOWECharDefault	EPM_Routines\Generic	Replace NULL or Dash characters to OWE Default Value (SPACE)
NullToOWEDateDefault	EPM_Routines\Generic	Replace NULL or MDW default dates(1753-01-01) to OWE Date Default Value (1900-01-01 00:00:00)
NullToSIDDefault	EPM_Routines\Generic	Replace NULL to Default SID Value
ReadParameterFile	EPM_Routines\Generic	Gets the value for the ParameterName from the Parameter file.
ResetJob	EPM_Routines\Generic	Resets the job
Round	EPM_Routines\Generic	Rounds numbers to a specified number of digits or decimal places and rounds one part of a date.
RtnBuildJobParam	EPM_Routines\Generic	Generates the JobParameters
RtnBuildParlMseq	EPM_Routines\Generic	Create the Master Sequence
RtnBuildSeqMseq	EPM_Routines\Generic	Create the Master Sequence
RtnGetEnvironmentVariable	EPM_Routines\Generic	Returns the Value for the Given Environment Variable
RunJob	EPM_Routines\Generic	Run a job

Routine Name	Category	Short Description
SbrtnDeleteVOC	EPM_Routines\Generic	Subroutine for deleting a VOC Entry
SbrtnSetVOC	EPM_Routines\Generic	Subroutine for Setting the VOC entry
TimeDiffSeconds	EPM_Routines\Generic	Compute the Time Difference in seconds between 2 timestamps.
ToChar	EPM_Routines\Generic	Converts a Date/Time data type to a string with the format specified.
ToCharn	EPM_Routines\Generic	Convert the number in string format to number
ToDate	EPM_Routines\Generic	Convert the format given to Date
ToInteger	EPM_Routines\Generic	Converts the input value to integer type.
ValidateHashLookup	EPM_Routines\Generic	Validates the Hashed File whether it is having default values or valid data
ValidateOWELkp	EPM_Routines\Generic	This routine will identify whether the input values are valid values or the default values
MCCBuildDateSQL	EPM_Routines\MDW_Currency_Conversion	Build SQL to get required date value for a given date granularity record name, DATE_DIM_REC.
MCCBuildSQL	EPM_Routines\MDW_Currency_Conversion	Build SQL for Currency Conversion
MCCE1RateCalc	EPM_Routines\MDW_Currency_Conversion	Currency Conversion Logic Using E1 Rate Table.
GenerateInputRH	EPM_Routines\Trees_RecursiveHierarchy	Code to write into a temp sequential file for RH
GetESourceNodeDescrLngSql	EPM_Routines\Trees_RecursiveHierarchy	Frame the SQL to get the Node Language Description(ESource Trees)
GetESourceNodeDescrSql	EPM_Routines\Trees_RecursiveHierarchy	Frame the SQL to get the Node Description(ESource Trees)
GetFlattenerEffdtClause	EPM_Routines\Trees_RecursiveHierarchy	Frame the Effdt subquery for the Flattener source DRS(EPM Trees).
GetNodeDescr	EPM_Routines\Trees_RecursiveHierarchy	Get the Node Descr for a given level
GetNodeDescrLng	EPM_Routines\Trees_RecursiveHierarchy	Get the Language Descr for a given level
GetNodeDescrLngSql	EPM_Routines\Trees_RecursiveHierarchy	Frame the SQL to get the Node language Description(EPM Trees).
GetNodeDescrSql	EPM_Routines\Trees_RecursiveHierarchy	Frame the SQL to get the Node Description(EPM Trees).
GetNodeID	EPM_Routines\Trees_RecursiveHierarchy	Get the Node ID for a given level.
GetParentInfo	EPM_Routines\Trees_RecursiveHierarchy	Get the Entity Information.
GetRecordNameDescr	EPM_Routines\Trees_RecursiveHierarchy	Get the Record Description
GetRhDnomParams	EPM_Routines\Trees_RecursiveHierarchy	Params for Recursive Hierarchy Denormalization
GetRhFlatParams	EPM_Routines\Trees_RecursiveHierarchy	Parameters for Recursive Hierarchy Flatenning
GetRHJobName	EPM_Routines\Trees_RecursiveHierarchy	Gets the Recursive Hierarchy Job Name to process.
GetStageRecname	EPM_Routines\Trees_RecursiveHierarchy	Gets the OWS record name for a given SRC record name
GetTreeDnomParams	EPM_Routines\Trees_RecursiveHierarchy	Parameters for Tree Denormalization
GetTreeFlatParams	EPM_Routines\Trees_RecursiveHierarchy	Parameters for Tree Flatenning
GetTreeFlatParamsLang	EPM_Routines\Trees_RecursiveHierarchy	Parameters for Tree Language jobs.
GetTreeJobName	EPM_Routines\Trees_RecursiveHierarchy	Gets the Tree Job Name to process.
GetTreeType	EPM_Routines\Trees_RecursiveHierarchy	Gets the Tree Type
InsertUVTable	EPM_Routines\Trees_RecursiveHierarchy	Insert Entity Information into UV Table.

Appendix D

Using the PeopleSoft EPM Lineage Spreadsheets

This document provides an overview of the EPM lineage spreadsheets and discusses how to use the spreadsheets to:

- View lineage information.
- Generate lineage information for a job.

Understanding the EPM Lineage Spreadsheets

The EPM lineage spreadsheets provide information about the ETL jobs that are delivered with the EPM warehouses. The spreadsheets act like a reverse-engineering tool or family tree; they enable you to view the ancestry of source, target, and lookup tables and their relevant ETL jobs. Each spreadsheet provides lineage information for a single warehouse. The following table lists the lineage spreadsheets that are currently available:

Lineage Spreadsheet Filename	*Warehouse*
ETL_CS_Lineage_Spreadsheet.xls	Campus Solutions Warehouse
ETL_CRM_Lineage_Spreadsheet.xls	CRM Warehouse
ETL_FMS_Lineage_Spreadsheet.xls	FMS Warehouse
ETL_HCM_Lineage_Spreadsheet.xls	HCM Warehouse
ETL_SCM_Lineage_Spreadsheet.xls	SCM Warehouse

By using the spreadsheets, you can:

- View lineage information for staging, dimension, and fact ETL jobs, or source, target, and lookup tables.
- Identify the sequence of jobs to run for a specific data mart.
- Identify inter-mart and cross-warehouse dependencies.
- Generate lineage information for a specific ETL job.

Spreadsheet Structure

Each EPM lineage spreadsheet includes several worksheets. The following table provides a description of each worksheet, by name, listed in the order in which it appears:

Worksheet	Description
Template	This worksheet contains overview information, a legend, and a definition of the columns used in the worksheets.
Setup	This worksheet contains ETL lineage information for all of the setup and staging jobs required for the warehouse.
Com Dims	This worksheet contains ETL lineage information for the common dimension jobs required for the warehouse.
Utils	This worksheet contains ETL lineage information for the currency conversion jobs required for the warehouse.
Global Dims	This worksheet contains ETL lineage information for the global dimension jobs required for the warehouse.
Local Dims	This worksheet contains ETL lineage information for the local dimension jobs required for the warehouse.
<*Data Mart*> For example: GL & Profitablity, ESA, Campus Community, and so on.	This worksheet contains ETL lineage information for the jobs required for a specific data mart. **Note.** Each spreadsheet includes several data mart worksheets.
Dynamic_Lineage_Generator	This worksheet provides a macro that enables you to enter the name of an ETL job and automatically generate a list of the complete lineage for that job.
JobOrder	This worksheet is an extension of the Dynamic_Lineage_Generator worksheet. It displays the order in which jobs need to be run.

Column Descriptions

The following table provides descriptions of the columns in the worksheets.

Column	Description
Sequencer Job	The name of the job sequencer, which is responsible for invoking and running other ETL server jobs.
Server Job	The name of the server job that is called by the job sequencer.

Column	*Description*
Server Job Category	The location of the server job in the IBM WebSphere DataStage project.
Target Table	The name of the target table used in the server job.
Target Update Action	The target load strategy for the server job.
Source Table	The name of the source table used in the server job.
Source Extraction Type	The type of extraction from the source table in the server job (for example, incremental date time or cyclical redundancy check).
Lookup Tables	The name of the lookup tables that are used in the server job. Lookups can be hashed files or direct DRS lookups. The lineage information captures the table names from which the hash files are populated and the table names for the direct DRS lookup.
Setup Jobs	The name of the setup job that populates the source and/or the lookup table.
Setup Sequencer Job	The name of the job sequencer that calls the setup server job.
MDW	The name of the MDW server job. This column has an entry if the source table or lookup table is populated from an MDW server job.
MDW Sequencer	The name of the MDW sequence job.
OWS	The name of the OWS server job. This column has an entry if the source table or lookup tables are populated from an OWS server job.
OWS Sequencer	The name of the OWS sequence job.
OWE	The name of the OWE server job. This column has an entry if the source table or lookup tables are populated from an OWE server job.
OWE Sequencer	The name of the OWE sequence job.
EPM Foundation	The application or EPM foundation setup page that populates the source table or the lookup table, such as Global Consolidations, Dimension Mapper, or setup PIA pages.
Category	The categories in which the setup jobs, MDW jobs, OWS jobs or OWE jobs are placed.

Column	*Description*
Comments	Any additional comments, if applicable.

Note. The spreadsheet does not contain lineage details for OWE jobs and Tree jobs, except for the GL&Profitability Mart of the FMS warehouse, which does include lineage information for OWE jobs.

Viewing Lineage Information

This section discusses how to use the spreadsheet to:

- Find lineage information for a server job.
- Identify the list of Jobs to be run for a data mart.

Finding Lineage Information for a Server Job

To find lineage information for a server job:

1. Access the worksheet in which the job is categorized.
2. Use Excel's Find feature to find the server job name in column B.
   a. Type Ctrl-F to access the Find and Replace Dialog box.
   b. Enter the name of the server job in the Find what edit box.
   c. Click Find Next until the job name is found in the Server Job column (column B).
   d. Close the Find dialog box.
3. Review the lineage information in the adjacent columns.

   The Sequencer Job column (column A) lists the sequencer which calls this job. The Server Job Category column (column C) lists the category this job is associated with. The Target Table, Target Update Action, Source Table, and Source Extraction Type for this server job are listed in columns D, E, F, and G respectively. The Lookup Tables column (Column H) lists all the lookups used by this job.

   The source tables and the lookup tables are placed in separate rows. This enables you to find the lineage information for each of these tables by navigating through the other subsequent columns in the same row. Columns I through R list the dependent jobs that are required to populate the source and lookup tables, and entries in these columns indicate whether the table is populated by Setup jobs, (column I), MDW jobs (column K), OWS jobs (column M), OWE jobs (column O), or Foundation setup / Apps (column Q). The Category column (column R) lists the category that the dependent job is associated with.

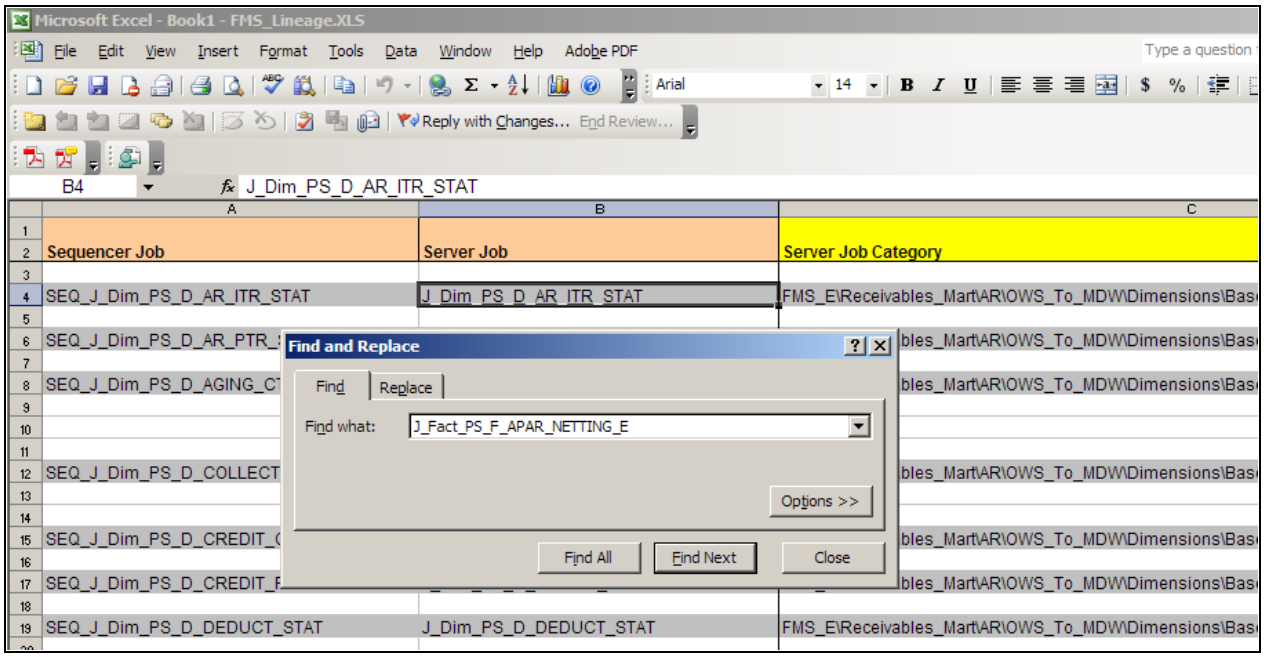
   Source tables that are from a different data mart (inter-mart) or different warehouse (cross-warehouse) are indicated by the colors specified in the legend on the Template worksheet page.

The spreadsheet lists the lineage of a source or lookup table to the level of the job that directly populates it. The lineage information does not extend to level of the last staging job. To get the complete lineage for a fact or dimension job fully extended through the lowest staging level, you can use the dynamic lineage generator tool, which generates a list of all the required dependent jobs that need to be run in order to load a particular fact or dimension.

Example

This example, from the ETL FMS Lineage spreadsheet, takes you through the tasks you would complete to review the information for the fact job J_Fact_PS_F_APAR_NETTING_E, which is used for the AR Data Mart.

1. Navigate to the AR worksheet page.
2. Type Ctrl-F and type J_Fact_PS_F_APAR_NETTING_E into the Find and Replace dialog box.



Find and Replace Dialog Box

3. Type Ctrl-F and type J_Fact_PS_F_APAR_NETTING_E into the Find and Replace dialog box.
4. Click Find Next until you access the cell in the Server Job column that contains the J_Fact_PS_F_APAR_NETTING_E job.

5. Close the Find and Replace dialog box. You should see the following information:

	A	B	
1			
2	Sequencer Job	Server Job	Server Job Category
90			
91	SEQ_Dims_L_O_CREDIT_CLASS	J_Dim_PS_O_CREDIT_CLASS	FMS_E\Receivables_Mart\AR\OWS_T
92			
93			
94	SEQ_Dims_L_O_CREDIT_RISK	J_Dim_PS_O_CREDIT_RISK	FMS_E\Receivables_Mart\AR\OWS_T
95			
96			
97	SEQ_Dims_L_O_DEDUCT_STAT	J_Dim_PS_O_DEDUCT_STAT	FMS_E\Receivables_Mart\AR\OWS_T
98			
99			
100	SEQ_Dims_L_O_DISPUTE_STAT	J_Dim_PS_O_DISPUTE_STAT	FMS_E\Receivables_Mart\AR\OWS_T
101			
102			
103	SEQ_Dims_L_O_ENTRY_RSTYP	J_Dim_PS_O_ENTRY_RSTYP	FMS_E\Receivables_Mart\AR\OWS_T
104			
105			
106	SEQ_J_Fact_PS_F_APAR_NETTING_E	J_Fact_PS_F_APAR_NETTING_E	FMS_E\Receivables_Mart\AR\OWS_T
107			
108			
109			
110			
111	SEQ_J_Fact_PS_F_AR_ACCOUNT_LN_E	J_Fact_PS_F_AR_ACCOUNT_LN_E_ITEM	FMS_E\Receivables_Mart\AR\OWS_T

J_Fact_PS_F_APAR_NETTING_E job displayed in spreadsheet

6. Scroll to the right to review the columns shown here:

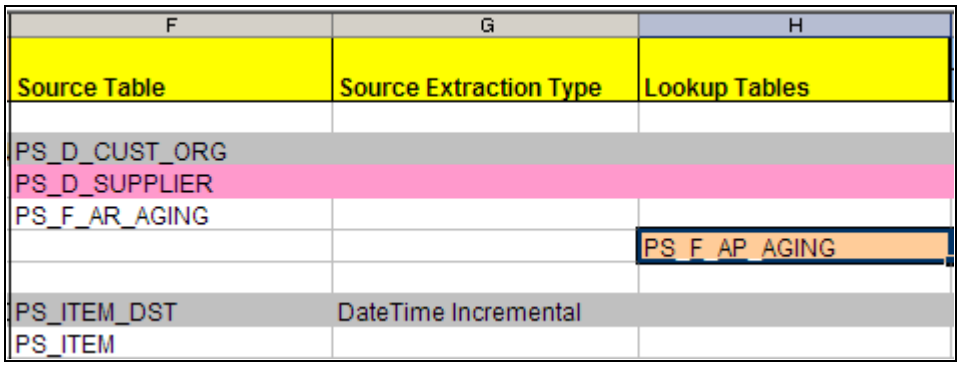
D	E	F	G
Target Table	Target Update Action	Source Table	Source Extraction Type
		PS_D_ENTRY_RSTYP	
PS_F_APAR_NETTING	Truncate table then insert rows	PS_D_CUST_ORG	
		PS_D_SUPPLIER	
		PS_F_AR_AGING	
PS_F_AR_ACCOUNT_LN	Insert new rows or update existing ones	PS_ITEM_DST	DateTime Incremental
		PS_ITEM	

Reviewing data associated with the J_Fact_PS_F_APAR_NETTING_E job

The Target Table, Target Update Action, Source Table, and Source Extraction Type for the J_Fact_PS_F_APAR_NETTING_E server job are listed in columns D, E, F, and G, respectively.

7. Continue to scroll to the right to view the remaining columns.

   The Lookup Tables column (Column H) lists all the lookups used in J_Fact_PS_F_APAR_NETTING_E.



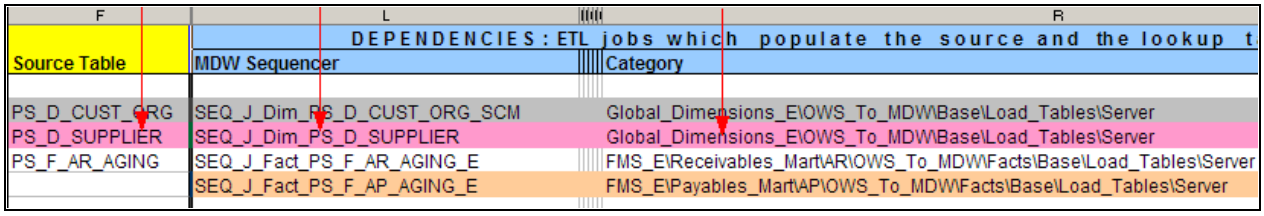
F	G	H
Source Table	Source Extraction Type	Lookup Tables
PS_D_CUST_ORG		
PS_D_SUPPLIER		
PS_F_AR_AGING		
		PS_F_AP_AGING
PS_ITEM_DST	DateTime Incremental	
PS_ITEM		

Lookup Tables Column

In this example there are three source tables: PS_D_CUST_ORG, PS_D_SUPPLIER, PS_F_AR_AGING. The lookup table is PS_F_AP_AGING. The source tables and the lookup tables are each placed in a unique row one after the other. This enables you to view the lineage information for each of these tables by navigating through the succeeding columns within the same row.

Columns I through R list out the dependent jobs required to populate these source and lookup tables. In this example, the source table PS_D_CUST_ORG has an entry in the MDW column, which means that is is populated from the MDW dimension J_Dim_PS_D_CUST_ORG_SCM, which is placed in the category Global_Dimensions_E\OWS_To_MDW\Base\Load_Tables\Server.

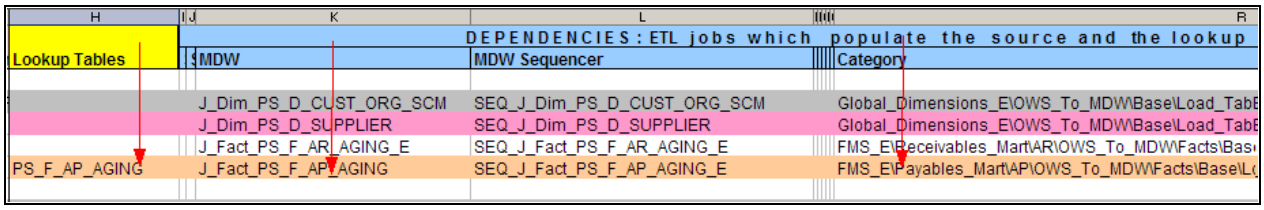
As shown in the following screenshot, the source table PS_D_SUPPLIER is an SCM warehouse dimension. The cross-warehouse dependency is identified by the different color (the color legend is located on the first worksheet page).



F	L	R
	DEPENDENCIES: ETL jobs which populate the source and the lookup t	
Source Table	MDW Sequencer	Category
PS_D_CUST_ORG	SEQ_J_Dim_PS_D_CUST_ORG_SCM	Global_Dimensions_E\OWS_To_MDW\Base\Load_Tables\Server
PS_D_SUPPLIER	SEQ_J_Dim_PS_D_SUPPLIER	Global_Dimensions_E\OWS_To_MDW\Base\Load_Tables\Server
PS_F_AR_AGING	SEQ_J_Fact_PS_F_AR_AGING_E	FMS_E\Receivables_Mart\AR\OWS_To_MDW\Facts\Base\Load_Tables\Server
	SEQ_J_Fact_PS_F_AP_AGING_E	FMS_E\Payables_Mart\AP\OWS_To_MDW\Facts\Base\Load_Tables\Server

Cross-warehouse dependencies for PS_D_SUPPLIER

Similarly, the lookup table PS_F_AP_AGING is populated from the fact job J_Fact_PS_F_AP_AGING placed in the category FMS_E\Payables_Mart\AP\OWS_To_MDW\Facts\Base\Load_Tables\Server. This fact job belongs to a different mart as indicated by the different color.



H	J	K	L	R
			DEPENDENCIES: ETL jobs which populate the source and the lookup	
Lookup Tables	MDW		MDW Sequencer	Category
	J_Dim_PS_D_CUST_ORG_SCM		SEQ_J_Dim_PS_D_CUST_ORG_SCM	Global_Dimensions_E\OWS_To_MDW\Base\Load_Tabl
	J_Dim_PS_D_SUPPLIER		SEQ_J_Dim_PS_D_SUPPLIER	Global_Dimensions_E\OWS_To_MDW\Base\Load_Tabl
	J_Fact_PS_F_AR_AGING_E		SEQ_J_Fact_PS_F_AR_AGING_E	FMS_E\Receivables_Mart\AR\OWS_To_MDW\Facts\Base
PS_F_AP_AGING	J_Fact_PS_F_AP_AGING		SEQ_J_Fact_PS_F_AP_AGING_E	FMS_E\Payables_Mart\AP\OWS_To_MDW\Facts\Base\L

Cross-warehouse dependencies for PS_F_AP_AGING

Identifying the List of Jobs to be Run for a Data Mart

You can use the information in the spreadsheet to identify the list of jobs that need to be run for a specific data mart. These include common jobs that are required for every data mart, which we refer to as prerequisite jobs, as well as jobs specific to the particular data mart.

If you prefer, you can create your own master sequencers based on the information provided in this section.

Alternatively, you can generate the list of jobs by using the Dynamic Lineage Generator tool. For more information, see "Generating Lineage Information for a Job".

Note. All the server jobs relating to Hash files that are present within the Load_Hash_Files category need to be run first before running other Sequence jobs within the Load_Tables category since these hash files are being used in other server jobs.

Prerequisite Jobs

The prerequisite jobs include setup jobs, staging jobs, and dimension jobs.

The following sets of jobs need to be run for *every* mart, in the order that they are listed in the worksheets:

1. Run these setup jobs in the Setup worksheet:

   a. All jobs within the Setup_E\OWS\<*Warehouse*> category.

      (For example all jobs within the Setup_E\OWS\FSCM category for the FMS warehouse and all jobs within the Setup_E\OWS\CS category for the CS warehouse).

   b. All jobs within the Setup_E\Dimension mapper category.

      Note. Please ensure that you run the Business Unit Wizard before proceeding with the following steps.

      See Chapter 14, "Importing Source Business Units into EPM to Create Warehouse Business Units," page 283.

   c. All jobs within the Shared_Lookups\DimensionMapper_Lookups category.

   d. All jobs within the Shared_Lookups\Control_Tables category.

   e. All jobs within the Shared_Lookups\System_Lookups category.

   f. All jobs within the Shared_Lookups\Language_Lookups category.

   g. All jobs within the Setup_E\OWE category (this step does not apply to the Campus Solutions warehouse).

   h. If you are implementing currency conversion, then run the jobs listed in the Utils worksheet.

2. Run the staging jobs listed in the OWS Sequencer column (column N) in the following worksheets:

   a. Com Dims.

   b. Global Dims.

   c. Local Dims.

   d. *<Data Mart>*, where *<Data Mart>* is the name of the data mart, for example AP, AR, Campus Community, Student Financials .

3. Run the Common Dimension Jobs listed in the Com Dims worksheet.

4. Run the Global Dimensions jobs listed in the Global Dims worksheet. (These jobs are required for running the FMS warehouse jobs.)

5. Run the Local Dimension Jobs placed in the Local Dims worksheet.

Data Mart Specific Jobs

Run all the Server jobs listed in column B of the worksheet for the specific data mart, to populate the corresponding Dimension and Fact tables for that mart.

Note. Do not run the jobs that are listed within the Reusable Jobs category. These jobs are not used to load target tables. They are automatically triggered by various Sequence jobs.

Generating Lineage Information for a Job

The Dynamic_Lineage_Generator worksheet contains a macro that generates a list of all the dependent jobs that are required for any ETL job. This will easily help you identify all the list of jobs to be run for a specific fact or dimension job.

To use the Dynamic Lineage Generator:

1. Access the Dynamic_Lineage_Generator worksheet.

2. Enter the job name in cell B1.

3. Click the Get Job Lineage button.

   The macro retrieves the lineage required for running this fact job from the setup, staging, and the dimension jobs and displays it in the cells below. The macro also copies the entire list of dependent jobs to the JobOrder worksheet, so you can identify the complete list to be run in sequence.

You must run the following prerequisite setup jobs before you run the jobs listed in the JobOrder worksheet:

- Setup_E\OWS\<*Warehouse Name*> Job Sequencer.

  For example Setup_E\OWS\FSCM Job Sequencer or Setup_E\OWS\CS Job Sequencer.

- Setup_E\Dimension mapper Job Sequencer.

- Run the Business Unit Wizard to populate the Dimension mapper tables.

  See Chapter 14, "Importing Source Business Units into EPM to Create Warehouse Business Units," page 283.

- Shared_Lookups\DimensionMapper_Lookups
- Shared_Lookups\Control_Tables
- Shared_Lookups\System_Lookups
- Shared_Lookups\Language_Lookups
- Setup_E\OWE Job Sequencer (this step does not apply to the Campus Solutions warehouse).

After you run the prerequisite setup jobs, then run the jobs listed in the JobOrder worksheet.

Index

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W

X