Informatica

Informatica® PowerExchange for PeopleSoft
10.5

User Guide for PowerCenter

Informatica PowerExchange for PeopleSoft User Guide for PowerCenter
10.5
March 2021

Publication Date: 2021-03-17

Table of Contents

Preface 7

Informatica Resources 7

- Informatica Network 7
- Informatica Knowledge Base 7
- Informatica Documentation 7
- Informatica Product Availability Matrices 8
- Informatica Velocity 8
- Informatica Marketplace 8
- Informatica Global Customer Support 8

Chapter 1: Configuring PowerExchange for PeopleSoft 9

Configuring PowerExchange for PeopleSoft Overview 9

- Connection Requirements 9

Step 1. Configure PeopleSoft Security 10

- Securing PeopleSoft Metadata 10
- Securing PeopleSoft Source Data 11

Step 2. Configure an ODBC Data Source 11

Chapter 2: Understanding PowerExchange for PeopleSoft 12

Understanding PowerExchange for PeopleSoft Overview 12

PeopleSoft Architecture 12

- PeopleSoft Security 13

Importing PeopleSoft Sources 13

- Creating Key Relationships 14

PeopleSoft Records 14

- Importing PeopleSoft Keys 14
- Importing Effective Dated Records 14

PeopleSoft Trees 15

- Tree Levels 16
- Detail Trees and the Detail Record 17
- Winter Trees 18
- Summary Trees 18

Flattening Trees 18

- Horizontally Flattening Trees 18
- Vertically Flattening Trees 19

Chapter 3: Working with PeopleSoft Sources 22

Working with PeopleSoft Sources Overview 22

- Organizing Definitions in the Navigator 22

Working with Records 23

Record Metadata. . . . 23
Viewing Records on the Records Tab. . . . 24
Viewing Records on the Panels Tab. . . . 24
Record Key Columns. . . . 25
Creating Customized Key Relationships. . . . 25
Working with Trees. . . . 26
Tree Metadata. . . . 26
Importing Trees. . . . 28
Creating Tree Source Definitions. . . . 29
Importing the Detail Record. . . . 30
Connecting to the PeopleSoft System. . . . 30
Entering Connection Information. . . . 31
Filtering Available Records and Trees. . . . 31
Importing PeopleSoft Source Definitions. . . . 31
Creating Tree Source Definitions. . . . 33
Step 1. Create a Tree Source Definition. . . . 33
Step 2. Import Tree Attributes. . . . 33
Editing PeopleSoft Source Definitions. . . . 34

Chapter 4: Application Source Qualifier for PeopleSoft Sources. . . . 36

Application Source Qualifier for PeopleSoft Sources Overview. . . . 36
Using Table Names for PeopleSoft Records. . . . 37
Using Parameters and Variables with PeopleSoft Sources. . . . 38
Understanding the Default Query. . . . 38
Editing the Default Query. . . . 39
Viewing the Default Query. . . . 39
Joining Source Data. . . . 39
Understanding the Default Join. . . . 40
Joining Records. . . . 40
Joining Detail Trees and Detail Records. . . . 41
Joining Detail Trees with Non-Detail Records. . . . 43
Entering a Source Filter. . . . 44
Filter Syntax. . . . 44
Creating a Source Filter. . . . 45
Validating Filter Syntax. . . . 45
Entering a Join Override. . . . 46
Validating Join Override Syntax. . . . 46
Using an Extract Override. . . . 47
Creating an Extract Override. . . . 48
Sorting Ports. . . . 49
Selecting Distinct Values. . . . 49
Selecting Current Rows. . . . 50
Linking the TO_EFFDT Port. . . . 50

Joining Effective Dated PeopleSoft Records. 51
Handling EFFSEQ when Extracting Current Rows. 53
Examples of Joining Effective Dated Records. 53
Configuring an Application Source Qualifier. 58

Chapter 5: Accessing XLATTABLE Data. 60

Accessing XLATTABLE Data Overview. 60
Locating Accurate Values. 60
Sourcing XLATTABLE Data. 61
Step 1. Import the XLATTABLE Table from PeopleSoft. 62
Step 2. Add the XLATTABLE Source Definition to the Mapping. 62
Step 3. Override the Default Query. 62
Step 4. Configure the Application Connection and Run Workflows. 63
Looking Up XLATTABLE Data. 63
Step 1. Import the XLATTABLE Table from the Database. 63
Step 2. Create and Configure the Lookup Transformation. 64
Step 3. Configure a PeopleSoft Application Connection. 67
Step 4. Enter Location Information in the Mapping or Session 67

Chapter 6: Creating PeopleSoft Sessions and Workflows. 68

Configuring a Session for PeopleSoft Sources. 68
Configuring a Session with PeopleSoft-only Sources. 68
Configuring a Session with Heterogeneous Sources. 69
Entering a Source Table Owner Name. 69
Configuring a PeopleSoft Session to Partition Data. 69
Scheduling a Workflow. 69

Appendix A: Datatype Reference 70

PeopleSoft and Transformation Datatypes. 70
Unsupported Datatypes. 71
PeopleSoft Datatypes. 71
PeopleSoft SubRecords. 71

Appendix B: PeopleSoft Language Codes 72

PeopleSoft Language Codes. 72

Appendix C: Glossary. 74

Index. 77

Preface

Use the *Informatica® PowerExchange® for PeopleSoft User Guide* to learn how to extract from and load to a PeopleSoft system by using PowerCenter Client. Learn to create a PeopleSoft connection, develop mappings, and run sessions in an Informatica domain.

Informatica Resources

Informatica provides you with a range of product resources through the Informatica Network and other online portals. Use the resources to get the most from your Informatica products and solutions and to learn from other Informatica users and subject matter experts.

Informatica Network

The Informatica Network is the gateway to many resources, including the Informatica Knowledge Base and Informatica Global Customer Support. To enter the Informatica Network, visit https://network.informatica.com.

As an Informatica Network member, you have the following options:

- Search the Knowledge Base for product resources.
- View product availability information.
- Create and review your support cases.
- Find your local Informatica User Group Network and collaborate with your peers.

Informatica Knowledge Base

Use the Informatica Knowledge Base to find product resources such as how-to articles, best practices, video tutorials, and answers to frequently asked questions.

To search the Knowledge Base, visit https://search.informatica.com. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team at KB_Feedback@informatica.com.

Informatica Documentation

Use the Informatica Documentation Portal to explore an extensive library of documentation for current and recent product releases. To explore the Documentation Portal, visit https://docs.informatica.com.

If you have questions, comments, or ideas about the product documentation, contact the Informatica Documentation team at infa_documentation@informatica.com.

Informatica Product Availability Matrices

Product Availability Matrices (PAMs) indicate the versions of the operating systems, databases, and types of data sources and targets that a product release supports. You can browse the Informatica PAMs at https://network.informatica.com/community/informatica-network/product-availability-matrices.

Informatica Velocity

Informatica Velocity is a collection of tips and best practices developed by Informatica Professional Services and based on real-world experiences from hundreds of data management projects. Informatica Velocity represents the collective knowledge of Informatica consultants who work with organizations around the world to plan, develop, deploy, and maintain successful data management solutions.

You can find Informatica Velocity resources at http://velocity.informatica.com. If you have questions, comments, or ideas about Informatica Velocity, contact Informatica Professional Services at ips@informatica.com.

Informatica Marketplace

The Informatica Marketplace is a forum where you can find solutions that extend and enhance your Informatica implementations. Leverage any of the hundreds of solutions from Informatica developers and partners on the Marketplace to improve your productivity and speed up time to implementation on your projects. You can find the Informatica Marketplace at https://marketplace.informatica.com.

Informatica Global Customer Support

You can contact a Global Support Center by telephone or through the Informatica Network.

To find your local Informatica Global Customer Support telephone number, visit the Informatica website at the following link:
https://www.informatica.com/services-and-training/customer-success-services/contact-us.html.

To find online support resources on the Informatica Network, visit https://network.informatica.com and select the eSupport option.

CHAPTER 1

Configuring PowerExchange for PeopleSoft

This chapter includes the following topics:

- Configuring PowerExchange for PeopleSoft Overview, 9
- Step 1. Configure PeopleSoft Security, 10
- Step 2. Configure an ODBC Data Source, 11

Configuring PowerExchange for PeopleSoft Overview

Before configuring PowerExchange for PeopleSoft, you need to install and configure PowerCenter and one or more supported versions of PeopleSoft. The administrators for each of these systems should perform the installation and configuration tasks for their respective systems.

To configure PowerExchange for PeopleSoft, complete the following steps:

1. Configure PeopleSoft security.
2. Configure an ODBC data source.

Connection Requirements

To connect to PeopleSoft with the PowerCenter Client and PowerCenter Integration Service, you need the following information:

- **A database user name and password for PeopleSoft metadata tables.** The user name must have SELECT permission to read metadata from PeopleSoft metadata tables.
- **A database user name and password for the underlying physical database tables for the PeopleSoft system.** The user name must have SELECT permission to read source data from physical tables in the underlying database of the PeopleSoft system. This user name can be the same as the database user name above.

- **Native connect string for the underlying database.** To import PeopleSoft metadata, configure an ODBC data source using the native connect string for the underlying database. When importing PeopleSoft source definitions, the Designer uses the ODBC data source and a database user name and password with SELECT permission for the PeopleSoft metadata tables to connect to the PeopleSoft metadata tables.

  To extract source data, you configure an application connection for a PeopleSoft database in the Workflow Manager. When you configure the application connection, you use the native connect string and a database user name and password with SELECT permission for the physical tables on the underlying database. The PowerCenter Integration Service uses the database connection to extract source data from underlying physical database tables.

Step 1. Configure PeopleSoft Security

You can extract data from a PeopleSoft system while protecting the PeopleSoft metadata and data. To access PeopleSoft metadata and data, the PowerCenter Client and PowerCenter Integration Service require a database user name and password to read both the PeopleSoft metadata tables and the underlying physical database tables. While you can use the table owner names as the database user name, you can keep PeopleSoft metadata and source data secure by creating new database user names with read-only access. You can create one user for metadata and one user for source data. Alternatively, you can create a single user to access both metadata and source data.

Securing PeopleSoft Metadata

To protect PeopleSoft metadata, create a new database user and grant the user SELECT permission for PeopleSoft metadata tables. PeopleSoft system table names begin with PS, such as PSRECDEFN.

Enter this database user name in the Import from PeopleSoft dialog box when you use the Designer to import PeopleSoft source definitions. Since this user name is not the owner of the PeopleSoft system tables, you also enter the table owner name to import PeopleSoft source definitions.

To allow the Designer to import PeopleSoft source definitions, the database user requires SELECT permission on the following PeopleSoft metadata tables:

- PSDBFIELD
- PSDBFIELDLANG
- PSDBFLDLABL
- PSDBFLDLABLLANG
- PSLOCK
- PSMENUDEFN
- PSMENUDEFNLANG
- PSMENUITEM
- PSMENUITEMLANG
- PSOPTIONS
- PSPNLDEFN
- PSPNLFIELD
- PSPNLGROUP
- PSPNLGROUPLANG

- PSRECDEFN
- PSRECDEFNLANG
- PSRECFIELD
- PSSTATUS
- PSTREEDEFN
- PSTREEDEFNLANG
- PSTREELEAF
- PSTREELEVEL
- PSTREENODE
- PSTREESTRCT

Note: Not all versions of PeopleSoft use the listed tables.

You can alternatively grant SELECT permission to *all* PeopleSoft metadata tables to enable importing PeopleSoft source definitions.

Securing PeopleSoft Source Data

To protect PeopleSoft source data, create a database user and grant the user SELECT permission for the underlying physical database tables containing source data. Underlying physical table names begin with PS_, such as PS_LEDGER. To access the XLATTABLE table as a source, grant the user SELECT permission on the XLATTABLE table as well.

Use this database user name to configure the connection object in the Workflow Manager. The PowerCenter Integration Service uses this connection object to extract PeopleSoft source data.

When you configure a session to extract PeopleSoft source data, enter the owner name as the source table name prefix to access source data.

Step 2. Configure an ODBC Data Source

To import a PeopleSoft source definition, create an ODBC data source for each PeopleSoft system you want to access. The PowerCenter Designer uses ODBC to connect to the PeopleSoft metadata tables to import PeopleSoft metadata.

When creating an ODBC data source, configure the data source to connect to the underlying database for the PeopleSoft system. For example, if the PeopleSoft system resides on an Oracle database, configure an ODBC data source to connect to the Oracle database.

CHAPTER 2

Understanding PowerExchange for PeopleSoft

This chapter includes the following topics:

- Understanding PowerExchange for PeopleSoft Overview, 12
- PeopleSoft Architecture, 12
- Importing PeopleSoft Sources, 13
- PeopleSoft Records, 14
- PeopleSoft Trees, 15
- Flattening Trees, 18

Understanding PowerExchange for PeopleSoft Overview

PowerExchange for PeopleSoft extracts data from PeopleSoft systems. To extract data from a PeopleSoft system, complete the following steps:

1. Define connection information for the PowerCenter Client and PowerCenter Integration Service to connect to the PeopleSoft system.
2. Import a source definition.
3. Create a mapping.
4. Create a session and workflow.

Although the overall steps for extracting from PeopleSoft are similar to extracting from other relational databases, completing them can differ. For example, to use an Oracle source definition in a mapping, you must connect it to a Source Qualifier transformation. To use a PeopleSoft source definition, you connect it to an Application Source Qualifier transformation.

This chapter provides an overview of the interactions between the PowerCenter and PeopleSoft systems.

PeopleSoft Architecture

PowerExchange for PeopleSoft supports extraction from PeopleSoft systems.

PeopleSoft saves metadata in metadata tables. PeopleSoft metadata provides a description and logical view of data stored in underlying physical database tables. It can include information such as the datatype or physical location of columns.

To import a PeopleSoft source definition, the Designer uses ODBC to connect to the PeopleSoft metadata tables and import PeopleSoft metadata.

PeopleSoft stores data in underlying physical database tables and uses SQL to communicate with the database server.

The PowerCenter Integration Service runs directly against the underlying physical database tables to extract data. The PowerCenter Integration Service uses the database connection code page to perform code page conversions.

PeopleSoft lets you keep language-sensitive metadata and data in multiple languages. Use PeopleSoft language codes to import language-sensitive metadata into the PowerCenter repository. You can also use PeopleSoft language codes to extract language-sensitive data.

PeopleSoft Security

You can extract data from a PeopleSoft system without compromising existing PeopleSoft security.

To access PeopleSoft metadata and data, the PowerCenter Client and PowerCenter Integration Service require a database username and password to read PeopleSoft metadata tables and underlying physical database tables. While you can use the table owner names as the database username, you can keep PeopleSoft metadata and source data secure by creating new database usernames with read-only access.

You can create two users: one for metadata analysis and one for source data extraction. Alternatively, you can create a single user to access both metadata and application data.

Related Topics:

- "Step 1. Configure PeopleSoft Security" on page 10

Importing PeopleSoft Sources

Before extracting data from a source, you need to import a source definition. Use the Designer to connect to the PeopleSoft metadata tables for PeopleSoft source metadata.

PowerExchange for PeopleSoft extracts source data from two types of PeopleSoft objects:

- Records
- Trees

You cannot configure some metadata imported from PeopleSoft records in the Designer. Therefore, always import PeopleSoft records from the Sources > Import from PeopleSoft menu in the Source Analyzer.

Importing trees is different than importing records. The PowerCenter Integration Service needs to denormalize trees to extract data from them. Depending on how you want to denormalize a tree, you can choose to import or create a source definition for a tree.

When you use a PeopleSoft source definition in a mapping, you connect it to an Application Source Qualifier transformation. Use the Application Source Qualifier to define the record set you want from PeopleSoft sources by creating a filter or extract override. You can also use the Application Source Qualifier to join data from related sources.

Creating Key Relationships

After you import PeopleSoft record definitions, you can create primary-foreign key relationships between them in the Source Analyzer. The primary-foreign key relationships exist in the metadata only. You can join PeopleSoft records in the Application Source Qualifier if the records have primary-foreign key relationships.

PeopleSoft Records

A PeopleSoft record is a table-like logical structure. Like a relational table, a record can contain columns with defined datatypes, precision, scale, and keys.

PowerCenter supports extracting data from the following PeopleSoft records:

- **SQL table.** Has a one-to-one relationship with an underlying physical database table. Key columns in SQL tables are unique.
- **SQL view.** Like a database view, SQL views offer an alternative view of information in one or more database tables. Key columns in SQL views can contain duplicate values.

Metadata related to each record is saved in the PeopleSoft metadata tables. Data for each record is saved in underlying database tables. By default, PeopleSoft names the underlying database tables after the record, PS_*Record_Name*. For example, data for the PeopleSoft record AE_REQUEST is saved in the PS_AE_REQUEST database table.

When you import a PeopleSoft record, the Designer imports both the PeopleSoft source name and the underlying database table name. By default, the Designer uses the PeopleSoft source name as the name of the source definition. The PowerCenter Integration Service uses the underlying database table name to extract source data.

When you import a PeopleSoft record, the Designer creates a source definition containing a column for each column in the record. The Designer also imports key information and effective date information.

Importing PeopleSoft Keys

PeopleSoft systems allow key columns that can contain unique values or duplicate values. The Designer uses the PeopleSoft key type to represent *both* types of key columns. When you import a source definition for a PeopleSoft SQL table record or view record into a PowerCenter repository, the Designer creates PeopleSoft key columns for both SQL table and view key columns.

Related Topics:

- "Record Key Columns" on page 25
- "Creating Customized Key Relationships" on page 25

Importing Effective Dated Records

PeopleSoft uses effective dates in records to maintain data history. Using the effective date record column, EFFDT, you can store future, current, and past data. You can store multiple occurrences of data based on when it goes into effect.

Some PeopleSoft records also use an effective sequence column, EFFSEQ, to track changes. This column is used in conjunction with effective date columns when an effective date alone is not a sufficient indicator of the most recent data.

When you import an effective dated PeopleSoft record, the Designer adds a new column to the source definition, TO_EFFDT. You can link the TO_EFFDT field in the Application Source Qualifier to other transformations and targets.

By default, the PowerCenter Integration Service extracts all data from a PeopleSoft record. However, if a record contains effective date or effective sequence columns, you can perform the following tasks in the Application Source Qualifier to filter source data:

- Specify to extract current rows for the default query.
- Enter a user-defined filter.
- Enter a user-defined extract override.
- Enter a user-defined join.

Related Topics:

- "Linking the TO_EFFDT Port" on page 50
- "Joining Effective Dated PeopleSoft Records" on page 51

PeopleSoft Trees

A PeopleSoft tree is an object that defines the groupings and hierarchical relationships between the values of a database field. A tree specifies how the PeopleSoft system groups the values of a database field for purposes of reporting or security access.

PeopleSoft uses levels to define and organize the relationship between different nodes. PowerExchange for PeopleSoft extracts data from trees with either of the following characteristics:

- Strictly enforced levels
- Loosely enforced levels

PowerExchange for PeopleSoft extracts data from the following PeopleSoft tree structure types:

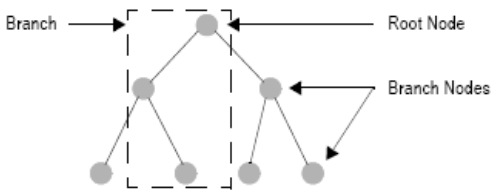
- **Detail trees.** Extracts data from loose-level and strict-level detail trees with static detail ranges.
- **Winter trees.** Extracts data from loose-level and strict-level node-oriented trees. Winter trees contain no detail ranges or detail records.
- **Summary trees.** Extracts data from loose-level and strict-level summary trees. Summary trees provide an alternate view of the nodes in a detail tree.

Each individual element in the tree is called a node. A tree can have two types of nodes:

- **Root.** The first level of the tree. All other nodes in the tree branch from the root node.
- **Branch.** Nodes extending from the root node containing information organized by the tree.

A logical unit of nodes extending from the root node level to the final branch node level of a tree is known as a *branch*.

The following figure represents a simple tree structure:



In addition, detail trees have *detail ranges*. Detail ranges represent the data in a related PeopleSoft record that is organized by the tree. The PeopleSoft record is the *detail record* for the tree.

Tree Levels

In PeopleSoft, you can use levels in a tree. PeopleSoft uses levels to define and organize the relationship between different nodes. The use of levels in a PeopleSoft tree can either be strictly enforced, loosely enforced, or not used.

PowerExchange for PeopleSoft extracts data from the following trees:

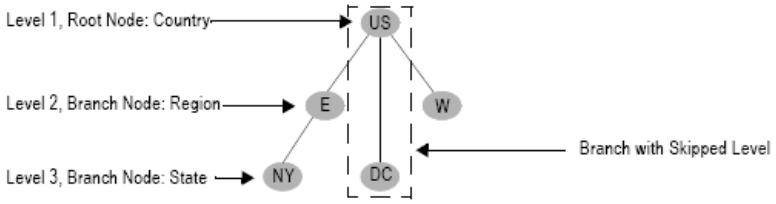
- **Strict-level trees.** A PeopleSoft tree with strictly enforced levels. All nodes at a particular level represent the same kind of information.
- **Loose-level trees.** A PeopleSoft tree with loosely enforced levels. Each level can contain nodes with different kinds of information.

Strict-Level Trees

PowerExchange for PeopleSoft supports trees with strictly enforced levels. In a strict-level tree, each branch of the tree progresses logically through the hierarchy of levels from the root node to the final branch node of a winter tree or the detail range of a detail tree. PeopleSoft assigns a level to each node based on its position in the tree.

Strict-level trees can contain branches with skipped or missing nodes. Skipped or missing nodes result in different sets of data but do not affect the structure of the tree or the imported source definition.

For example, "PeopleSoft Trees" on page 15 has no skipped or missing nodes. The following figure is a strict-level tree with missing nodes:



The tree contains defined levels, and all branches progress logically through the hierarchy of levels. The top level, or root node level, is the country US. Level 2 is a branch node level with the East and West regions. Level 3 is another branch node level consisting of states in each region.

The US-DC branch skips the region node level. The tree also contains a branch, US-W, with a missing state node level. In this strict-level tree, branches cannot be configured to alter or reverse the hierarchy of levels. For example, you cannot have the state node above the country node, such as DC-US.

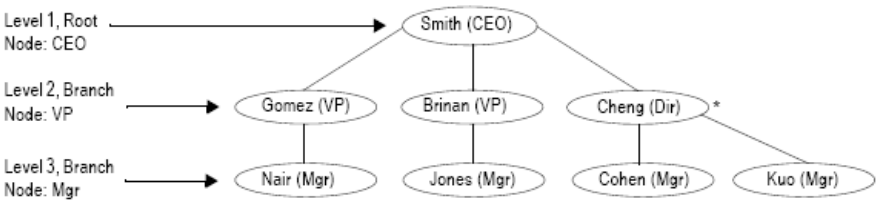
The PowerCenter Integration Service denormalizes strict-level trees using either vertical or horizontal flattening.

Loose-Level Trees

PowerExchange for PeopleSoft supports trees with loosely enforced levels. In a loose-level tree, one level can contain nodes with different kinds of information. Also, nodes representing the same type of information can appear at multiple levels. Loose-level trees do not have strict hierarchical structures.

Loose-level trees can contain branches with skipped or missing nodes. Skipped or missing nodes result in different sets of data but do not affect the structure of the tree or the imported source definition.

The following figure shows a loose-level winter tree with no missing nodes:



** Represents the node that makes this a loose-level tree*

Level one contains CEO information. Level two contains information about vice presidents. Level three contains information about managers. However, in this example, one node in level two refers to a data column that contains information about directors, not information about vice presidents.

The tree is a loose-level tree because the nodes in level two contain different kinds of information. The PowerCenter Integration Service denormalizes loose-level trees using vertical flattening only.

Detail Trees and the Detail Record

In a detail tree, any node in the tree can have associated detail ranges. PeopleSoft detail ranges can be static or dynamic. PowerExchange for PeopleSoft supports importing detail trees with *static* detail ranges.

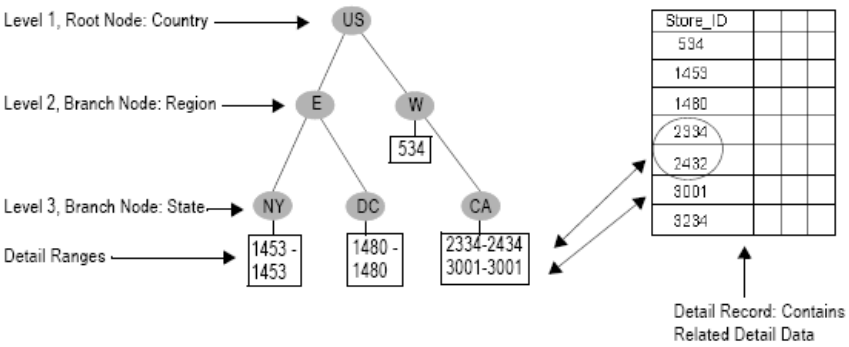
The detail ranges in a detail tree represent data in a column of a related PeopleSoft detail record. When a detail range contains a single value, it represents a single detail value in the detail record. When a detail range contains an upper and lower value, it represents values in the detail record that fall within the detail range.

Although detail ranges represent detail values in a detail record, they do not necessarily match detail values in the detail record. The numeric values defining a detail range can be values that do not exist in the detail record. Likewise, the detail record can contain values that are not represented in detail ranges.

For example, in the strict-level detail tree in the following figure, the detail range for the US-W-CA branch has detail ranges of 2334-2434 and 3001. These ranges correspond to values in the Store_ID column of the detail record.

The detail range 2334-2434 represents values in the detail record that fall within the range, 2334 and 2432. 2434 is not a value in the detail record even though it represents the upper value of the detail range. The 3001 detail range corresponds to the 3001 value in the detail record. The detail record contains a store ID value not represented by the tree, 3234.

The following figure shows the US-W branch that contains a missing node, but has an associated detail range:



Winter Trees

A winter tree is a node-oriented tree consisting of root and branch nodes only. Winter trees do not have associated detail ranges or detail records. The tree nodes in a winter tree represent the data values for a database field. "Strict-Level Trees" on page 16 is a winter tree. As with detail trees, winter trees can include skipped or missing nodes. However, winter trees can only include skipped or missing nodes if each branch in the tree conforms to the logical hierarchy of the tree.

Summary Trees

A summary tree provides an alternative view of nodes in a detail tree. When you create a summary tree in PeopleSoft, you specify the detail tree on which the summary tree is based.

Flattening Trees

When you extract data from a PeopleSoft tree, the PowerCenter Integration Service denormalizes the tree structure. The use of levels in the PeopleSoft tree determines which flattening methods you can use to denormalize the tree. The PowerCenter Integration Service denormalizes PeopleSoft trees using one of the following flattening methods:

- **Horizontal flattening.** The PowerCenter Integration Service creates a single row for each final branch node or detail range in the tree. You can only use horizontal flattening with strict-level trees. Horizontal flattening usually results in fewer columns for the tree source definition.
- **Vertical flattening.** The PowerCenter Integration Service creates a row for each node or detail range represented in the tree. Use vertical flattening with both strict-level and loose-level trees.

You can import or create PeopleSoft tree source definitions. If you want the PowerCenter Integration Service to horizontally flatten the tree, *import* the tree source. If you want the PowerCenter Integration Service to vertically flatten the tree, *create* the tree source.

The following table shows the flattening methods for different PeopleSoft tree level types and explains how to extract the PeopleSoft metadata for each tree level type:

Tree Levels	Flattening Method	Metadata Extraction Method
Strict-level tree	Horizontal	Import the source definition.
Strict-level tree	Vertical	Create the source definition.
Loose-level tree	Vertical	Create the source definition.

Horizontally Flattening Trees

The PowerCenter Integration Service uses horizontal flattening to denormalize strict-level trees that you *import* in the Source Analyzer. When you extract data from an imported tree with missing nodes, the PowerCenter Integration Service replaces missing nodes with NULL values.

Horizontally Flattening Winter Trees

When horizontally flattening a winter tree, the PowerCenter Integration Service creates a single row for each final branch node in the tree.

The following table shows an example of the data returned when the PowerCenter Integration Service uses horizontal flattening to extract data from the winter tree in "Strict-Level Trees" on page 16:

Country	Region	State
US	E	NY
US	NULL	DC
US	W	NULL

Horizontally Flattening Detail and Summary Trees

When horizontally flattening a detail tree, the PowerCenter Integration Service creates a single row for each detail range in the tree. Each row includes the logical node structure from the root node to the detail range.

The following table shows an example of the data returned when the PowerCenter Integration Service extracts data from the detail tree in "Detail Trees and the Detail Record" on page 17 using horizontal flattening:

Country	Region	State	RANGE_FROM	RANGE_TO
US	E	NY	1453	1453
US	E	DC	1480	1480
US	W	NULL	534	534
US	W	CA	2334	2434
US	W	CA	3001	3001

The PowerCenter Integration Service creates two rows containing US-W-CA, one for each detail range. The first three rows have RANGE_FROM and RANGE_TO populated with the same values because the detail ranges represent a single value. The US-W branch contains a null value in the State column to represent the missing State node.

The PowerCenter Integration Service horizontally flattens a summary tree the same way it flattens a detail tree.

Vertically Flattening Trees

The PowerCenter Integration Service uses vertical flattening to denormalize trees that you *create* in the Source Analyzer. The PowerCenter Integration Service creates a row for each node or detail tree represented in the tree. If the tree contains a missing node, the PowerCenter Integration Service does not create a row for it.

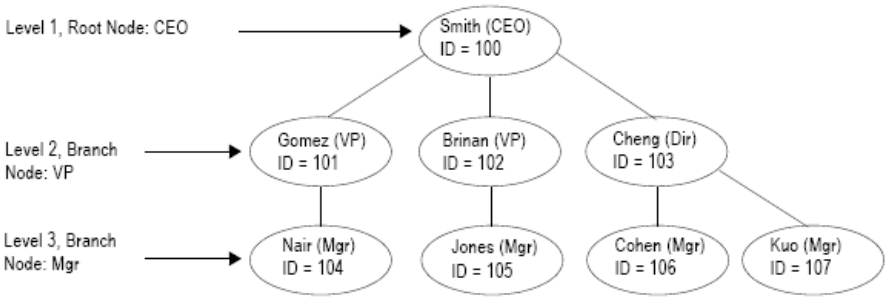
The tree source definition you create in the Source Analyzer is the same for all trees with the same columns. The PowerCenter Integration Service makes no distinction between strict-level and loose-level trees. The PowerCenter Integration Service makes no distinction between winter, detail, and summary trees. The RANGE_FROM and RANGE_TO columns are specific to detail trees. If the PowerCenter Integration Service vertically flattens a winter tree, it assigns NULL to the RANGE_FROM and RANGE_TO columns. Similarly, the

PowerCenter Integration Service assigns NULL to the RANGE_FROM and RANGE_TO columns for top-level nodes of a detail tree.

Vertically Flattening Winter Trees

When vertically flattening a winter tree, the PowerCenter Integration Service creates a single row for each node in the tree.

The following figure shows a tree similar to the tree in "Loose-Level Trees" on page 16, but it also shows each node ID:



The following table shows an example of the data returned when the PowerCenter Integration Service extracts data from the loose-level winter tree:

NODE_ID	PARENT_NODE_ID	CHILD_NODE_ID	SIBLING_NODE_ID	NODE_LABEL	PARENT_NODE_LABEL	CHILD_NODE_LABEL	SIBLING_NODE_LABEL	RANGE_FROM	RANGE_TO	NODE_LEVEL
100	NULL	101	NULL	Smith	NULL	Gomez	NULL	NULL	NULL	1
101	100	104	102	Gomez	Smith	Nair	Brinan	NULL	NULL	2
102	100	105	103	Brinan	Smith	Jones	Cheng	NULL	NULL	2
103	100	106	NULL	Cheng	Smith	Cohen	NULL	NULL	NULL	2
104	101	NULL	NULL	Nair	Gomez	NULL	NULL	NULL	NULL	3
105	102	NULL	NULL	Jones	Brinan	NULL	NULL	NULL	NULL	3
106	103	NULL	107	Cohen	Cheng	NULL	Kuo	NULL	NULL	3
107	103	NULL	NULL	Kuo	Cheng	NULL	NULL	NULL	NULL	3

When a node ID is NULL, its corresponding node label is also NULL. For example, for the first row of data, both PARENT_NODE_ID and PARENT_NODE_LABEL are NULL. The NODE_ID 100 is the root node and has no parent.

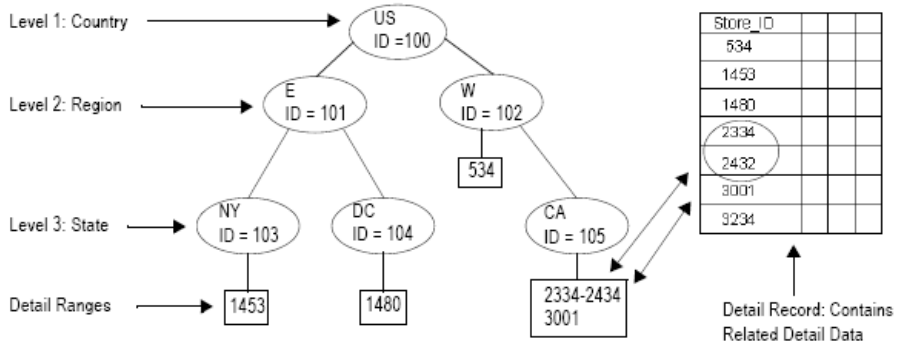
Level 2 contains three nodes, all of which are children to the root node. However, only two of the nodes specify a sibling node in the extracted data. The last node in the level does not specify a sibling node. You can determine its sibling relationship from the other sibling relationships defined. For example, for every parent node that has five children, the resulting data only specifies four sibling relationships.

Only two nodes in level 3 have sibling relationships with each other. NODE_ID 106 and NODE_ID 107 are siblings. However, NODE_ID 104 and NODE_ID 105 have no siblings because they have different parents than all other nodes in their level. The RANGE_FROM and RANGE_TO columns are all NULL because the tree source is a winter tree.

Vertically Flattening Detail and Summary Trees

When vertically flattening a detail tree, the PowerCenter Integration Service creates a single row for each node and detail range in the tree.

The following figure shows a tree similar to the tree in "Detail Trees and the Detail Record" on page 17, but it also shows each node ID:



The following table shows an example of the data returned when the PowerCenter Integration Service extracts data from the strict-level detail tree:

NODE_ID	PARENT_NODE_ID	CHILD_NODE_ID	SIBLING_NODE_ID	NODE_LABEL	PARENT_NODE_LABEL	CHILD_NODE_LABEL	SIBLING_NODE_LABEL	RANGE_FROM	RANGE_TO	NODE_LEVEL
100	NULL	101	NULL	US	NULL	E	NULL	NULL	NULL	1
101	100	103	102	E	US	NY	W	NULL	NULL	2
102	100	105	NULL	W	US	CA	NULL	534	534	2
103	101	NULL	104	NY	E	NULL	DC	1453	1453	3
104	101	NULL	NULL	DC	E	NULL	NULL	1480	1480	3
105	102	NULL	NULL	CA	W	NULL	NULL	2334	2432	3
105	102	NULL	NULL	CA	W	NULL	NULL	3001	3001	3

The extracted data contains two rows for node ID 105, one for each detail range.

The PowerCenter Integration Service vertically flattens a summary tree the same way it vertically flattens a detail tree.

CHAPTER 3

Working with PeopleSoft Sources

This chapter includes the following topics:

- Working with PeopleSoft Sources Overview, 22
- Working with Records, 23
- Working with Trees, 26
- Connecting to the PeopleSoft System, 30
- Importing PeopleSoft Source Definitions, 31
- Creating Tree Source Definitions, 33
- Editing PeopleSoft Source Definitions, 34

Working with PeopleSoft Sources Overview

Use the following PeopleSoft sources in a mapping:

- **Records.** Similar to a relational table. You can import PeopleSoft SQL table and SQL view records.
- **Trees.** An object describing the hierarchical relationships between values in a single column of the detail record. You can import detail trees, summary trees, and winter trees.

You can either import or create source definitions. You can only import PeopleSoft records. When importing a PeopleSoft source definition, the Designer imports PeopleSoft metadata that is not configurable in the Source Analyzer. Therefore, do not manually create source definitions for PeopleSoft records.

You can either import a PeopleSoft tree or create a tree source definition for a tree, depending on the tree type. You can only import strict-level trees. You can create a tree source definition for either strict-level or loose-level trees.

When you import an effective dated PeopleSoft record, you connect to the PeopleSoft system to access the PeopleSoft metadata tables. The Designer uses metadata in these tables to create the source definition. You cannot configure this metadata manually.

Organizing Definitions in the Navigator

After you import or create a PeopleSoft record or tree, the Navigator displays and organizes sources by the PeopleSoft record or tree name by default.

You can click Tools > Options to display business names for sources. The Designer displays sources by PeopleSoft business names and user-defined business names for created tree source definitions. Record and tree names appear in parentheses, adjacent to the business name.

Working with Records

When you import a PeopleSoft record, you connect to the PeopleSoft system to access the PeopleSoft metadata tables. The Designer uses metadata in these tables to create the source definition. You cannot configure this metadata manually.

You can import the following PeopleSoft record definitions into the repository:

- **SQL table.** PeopleSoft metadata similar to a relational table.
- **SQL view.** PeopleSoft metadata similar to a relational view.

You can import records from two tabs in the Import from PeopleSoft dialog box:

- **Records tab.** Records appear in alphabetical order on the Records tab.
- **Panels tab.** Records used by panels appear in the PeopleSoft organizational hierarchy on the Panels tab.

In the Import from PeopleSoft dialog box, the Designer displays the record name as it appears in the underlying database. Since the record name might differ from the PeopleSoft business name, the Import from PeopleSoft dialog box also displays the PeopleSoft business name for each record.

When you import a PeopleSoft record with an effective date, the Designer adds a new column to the source definition, TO_EFFDT. The TO_EFFDT column defines the end date for a range of valid dates.

Record Metadata

When you import a record as a source, the Designer imports column names and datatypes as it would any other relational source table. In addition, the Designer imports PeopleSoft record metadata. You can extend this record metadata by using metadata extensions.

The following table lists the PeopleSoft record metadata the Designer imports:

Source Definition Field	Contains
Source Name	Record name.
Table Type	PeopleSoft object type (record).
Business Name	PeopleSoft business name.
Attribute	IsLang. Indicates a column is language-sensitive and might have data available in different languages. The Designer only lists language-sensitive columns.
Extension Name	Metadata extension name.
Physical Table Name	Name of the physical database table containing source data. The field appears on the Attributes tab of the source definition when opened in the Mapping Designer.
Language Table Name	Name of the related language table containing language-sensitive data in different languages. The field appears on the Attributes tab of the source definition when opened in the Mapping Designer.

For example, in the following PeopleSoft source definition, the PeopleSoft object type, PeopleSoft Record, appears in the Table Type field. The PeopleSoft business name, Body Parts Codes, appears in the Business Name field.

On the Attributes tab, the Designer lists language-sensitive columns, such as DESCR100 and DESCRSHORT. It does not list columns that are not language sensitive.

On the Properties tab, the Designer displays the database table name for the record and the related language table. For example, the database table for the BODY_PART_TBL source is PS_BODY_PART_TBL, and the related language table is PS_BODY_PART_LANG. In this case, the related language table contains translated values for the language-sensitive fields, DESCR100 and DESCRSHORT.

On the Metadata Extensions tab, the Designer lists metadata extensions defined for the record. You use metadata extensions to extend the metadata stored in the repository by associating information with individual repository objects.

Viewing Records on the Records Tab

When you import a record, the Designer lists records in alphabetical order in the Records folder on the Records tab. Each record displays with the PeopleSoft business name.

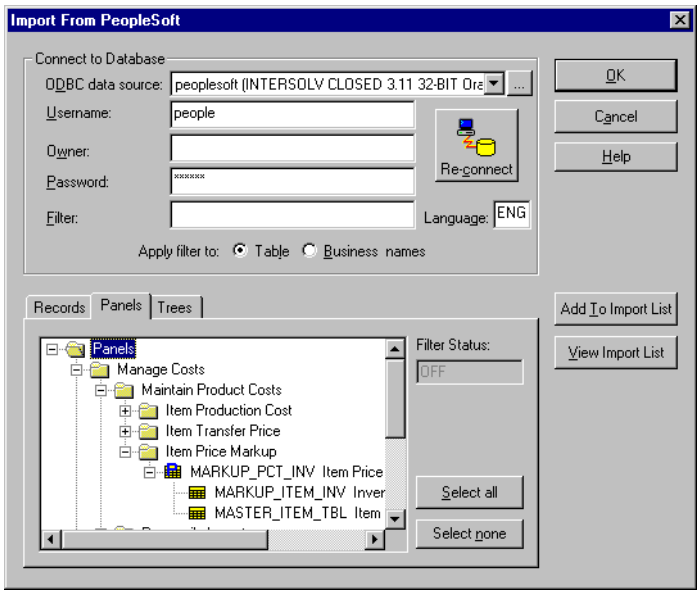
Viewing Records on the Panels Tab

You can import PeopleSoft records used by PeopleSoft panels from the Panels folder on the Panels tab. The Designer displays these records in the PeopleSoft organizational hierarchy. Expand the Panels folder to view the PeopleSoft navigational hierarchy. The Panels folder contains the following organizational hierarchy:

- Menu groups
- Menus
- Panel groups
- Panels
- Records

Only import records from the Panels tab.

The following figure shows the Panels tab in the Import from PeopleSoft dialog box:



The previous figure shows the following types of panels:

Panel Type	Example
Menu Group	Manage Costs
Menu	Maintain Product Costs
Panel Group	Item Transfer Price
Panels	MARKUP_PCT_INV Item Price
Records	MARKUP_ITEM_INV Inventory

The Panels tab displays panel groups under the PeopleSoft “Use” bar name and all other bar names.

Record Key Columns

When you import a PeopleSoft record with a key column, the Designer creates a PeopleSoft key. Since key columns from PeopleSoft SQL view sources often contain duplicate values, the Designer allows duplicate values in PeopleSoft key columns.

Columns in PeopleSoft sources are related if they have matching column names and at least one column is a PeopleSoft key column. The Designer uses connectors to display key relationships between imported records.

The columns are related because columns in ABSENCE_CAL have PeopleSoft key columns with matching column names in ABSENCE_CAL_VW.

The PowerCenter Integration Service uses the key relationships to create default joins in the Application Source Qualifier.

Creating Customized Key Relationships

You can join records in the Application Source Qualifier if the records have primary-foreign key relationships. You can create primary-foreign key relationships in the Source Analyzer. These columns do not have to be keys, but you can increase performance by including them in the index for each table.

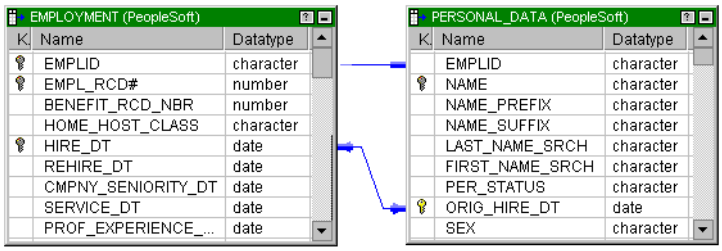
For example, you might want to join two records based on two date fields. The EMPLOYMENT and PERSONAL_DATA records have a key relationship based on EMPLID. Both records have hire date columns, EMPLOYMENT.HIRE_DT and PERSONAL_DATA.ORIG_HIRE_DT. However, the records do not share primary and foreign keys based on hire dates. Use the Source Analyzer to create a primary-foreign key relationship in the metadata.

The two records are not linked on the hire date columns. Therefore, the Designer does not recognize a relationship on the hire date columns.

You can create a relationship between the EMPLOYMENT and PERSONAL_DATA records in the Source Analyzer in the following ways:

- Drag a connection between the two columns.
- Edit the key type information for both source definitions.

The following figure shows the EMPLOYMENT and PERSONAL_DATA records after you defined a primary-foreign key relationship between the HIRE_DT and ORIG_HIRE_DT columns. The Designer uses an arrow to show the user-defined relationship:



The primary-foreign key relationships you define exist in the metadata only. You do not need to alter the source records. Once the key relationships exist, use an Application Source Qualifier to join the two tables. The default join includes a WHERE clause based on the two hire date columns.

To create a customized primary-foreign key relationship by editing the key types:

1. In the Source Analyzer, double-click the title bar of the source definition for which you want to define a primary key.
2. From the Columns tab, define the Key Type of the record column as either PRIMARY KEY or PRIMARY/PeopleSoft KEY, depending on the nature of the relationship with any other PeopleSoft records.
3. Click OK.
4. Double-click the title bar of the source definition for which you want to define a foreign key.
5. From the Columns tab, define the Key Type of the record column as FOREIGN KEY.
6. Select the table in which you defined the PRIMARY/PeopleSoft KEY in the Primary Table field.
7. Select the column you defined as the PRIMARY/PeopleSoft KEY in the Primary Column field.
8. Click OK.

The Designer shows a link between the two record source definitions with an arrow.

Working with Trees

You can import a PeopleSoft tree or create a tree source definition for a tree, depending on the tree type. You can only import strict-level trees. You can create a tree source definition for either strict-level or loose-level trees.

When you *create* a tree source definition, the Designer denormalizes the tree using vertical flattening. When you *import* a PeopleSoft tree, the Designer denormalizes the tree using horizontal flattening.

Tree Metadata

You can import tree metadata from the following PeopleSoft trees:

- Detail trees
- Summary trees
- Winter trees

You import PeopleSoft tree metadata at different times, depending on whether you import or create the tree source definition.

When you import a PeopleSoft tree source definition, you connect to the PeopleSoft system to access the PeopleSoft metadata tables. The Designer uses metadata in these tables to create the source definition in the Source Analyzer. You cannot configure this metadata manually.

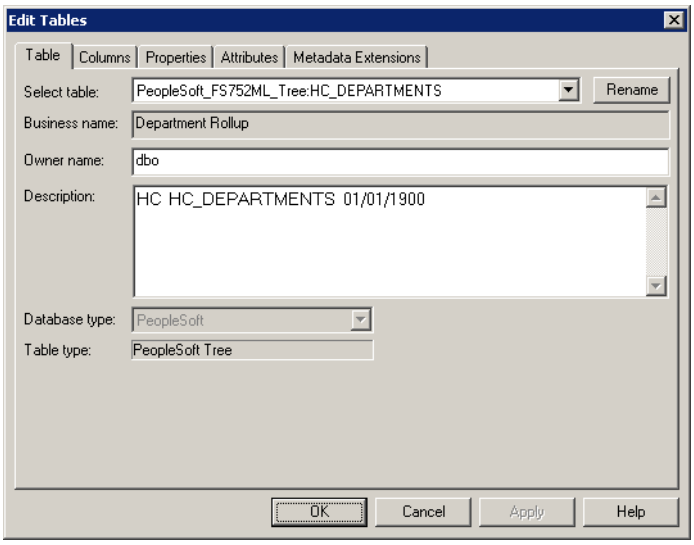
When you create a tree source definition, the Designer creates a source definition based on a predefined tree structure stored in the repository. Therefore, you do not see or import PeopleSoft tree metadata for the created tree source definition in the Source Analyzer. To associate PeopleSoft tree metadata with a created tree source definition, you import Peoplesoft tree attributes into the created tree source definition in the Mapping Designer.

The following table shows the PeopleSoft metadata the Designer imports when you import a tree in the Source Analyzer or when you import PeopleSoft tree attributes into a created tree source definition in the Mapping Designer:

Source Definition Field	Contains	Applicable Tree Source Definitions
Source Name	Tree name.	Created and imported
Business Name	Business name for the tree.	Imported
Owner Name	Owner of the tree.	Created and imported
Table Type	PeopleSoft object type (PeopleSoft tree, PeopleSoft winter tree, or PeopleSoft tree for vertical flattening).	Created and imported
SetID	A value that provides a unique identifier for each hierarchy. It is used to organize and identify data, such as MFG for Manufacturing. Each PeopleSoft tree has a SetID.	Created and imported
Effective Date	Effective date for the tree.	Created and imported
Tree Name	Name of the tree.	Created and imported
Detail Table Name	Detail record name. The field appears on the Attributes tab of the source definition when opened in the Mapping Designer.	Imported
Detail Field Name	Column in the detail record used to join the detail record with the detail tree. The field appears on the Attributes tab of the source definition when opened in the Mapping Designer.	Imported
Set Control Value	Additional identifier used by some versions of PeopleSoft to identify winter trees. The field appears on the Attributes tab of the source definition when opened in the Mapping Designer.	Created and imported

Note: To add or view metadata for a created PeopleSoft tree source definition, open the Edit Transformations dialog box for the source definition in the Mapping Designer.

The following figure shows an imported source definition:



The PeopleSoft object type and PeopleSoft Tree appear in the Table Type field. The SetID (HC), tree name (HC_DEPARTMENTS), and effective date (01/01/1900) appear in the Description field. The PeopleSoft business name, Department Rollup, appears in the Business Name field.

Importing Trees

You import strict-level trees from the Trees tab of the Import from PeopleSoft dialog box. Detail and summary trees appear in the Trees folder. Winter trees appear in the Winter Tree folder.

The Designer displays the following tree information in the Import from PeopleSoft dialog box to help you identify the tree you want to import:

- **SetID.** If a tree has a SetID, it appears before the tree name.
- **Tree name.** The name of the tree.
- **Effective date.** The tree effective date appears after the tree name.

PeopleSoft uses SetID and effective dates to identify trees. When importing a tree from PeopleSoft, use the SetID and effective date to select the tree. The SetID and effective date appear in the resulting source definition in the Source Analyzer.

PowerExchange for PeopleSoft only *imports* strict-level trees. When you import a strict-level tree, the PowerCenter Integration Service uses horizontal flattening to denormalize the tree.

Note: To use a loose-level tree as a source, you must create the source definition.

When you import strict-level trees, the Designer creates columns for the following:

- **Root node level.** The Designer creates single column for the root node level and names it LEVEL_*RootNodeName*.
- **Branch node levels.** The Designer creates a column for each branch node level in the tree and names each column LEVEL_*BranchNodeLevel*.
- **Detail ranges.** The Designer creates columns named RANGE_FROM and RANGE_TO to represent the detail ranges in detail and summary trees. Winter trees do not have detail ranges.

All columns are Character datatypes with a precision of 30.

Creating Tree Source Definitions

You can create a tree source definition in the Source Analyzer. However, when you create a tree source definition, the PowerCenter Integration Service always uses vertical flattening to denormalize the tree.

To use a loose-level PeopleSoft tree as a source in a mapping, you must create the source definition. However, you can create tree source definitions for either loose-level or strict-level trees. You cannot import loose-level trees.

When you create a tree source definition, the Designer uses a predefined source in the repository that specifies the column definitions. You cannot manually edit the tree source definition.

The following table lists the columns in the created tree source definition and their descriptions:

Column Name	Datatype	Length	Description
NODE_ID	Number	10	Unique node ID of the tree.
PARENT_NODE_ID	Number	10	Node ID of the parent node for the node represented by NODE_ID.
CHILD_NODE_ID	Number	10	Node ID of the child node for the node represented by NODE_ID.
SIBLING_ID	Number	10	Node ID of the sibling node for the node represented by NODE_ID.
NODE_LABEL	Character	30	Node label of the node represented by NODE_ID.
PARENT_NODE_LABEL	Character	30	Node label of the parent node for the node represented by NODE_ID.
CHILD_NODE_LABEL	Character	30	Node label of the child node for the node represented by NODE_ID.
SIBLING_NODE_LABEL	Character	30	Node label of the sibling node for the node represented by NODE_ID.
RANGE_FROM	Character	30	Range from value of the node.
RANGE_TO	Character	30	Range to value of the node.
NODE_LEVEL	Number	10	Level number of the node represented by NODE_ID.

You only need to create one tree source definition in the Source Analyzer. You can use the source definition multiple times in a mapping. However, you need to associate a particular PeopleSoft tree with each instance in a mapping. To associate a PeopleSoft tree with each instance in a mapping, import the attributes of the tree into each *instance* of the tree source definition.

Use the created tree source definition in a mapping for all PeopleSoft tree types. When denormalizing a PeopleSoft tree from a created tree source definition, the PowerCenter Integration Service makes no distinction between winter, detail, and summary trees.

Importing Tree Attributes

To import PeopleSoft tree attributes, use the Import Attributes from PeopleSoft dialog box in the Mapping Designer. You import PeopleSoft tree attributes from the Trees tab of the Import from PeopleSoft dialog box.

All available loose-level and strict-level trees appear in the Trees tab. There is only one folder, the Trees folder, where the Designer lists all trees. The Designer does not distinguish between winter, detail, and summary trees when you import tree attributes into created tree source definitions.

The Designer displays the following tree information in the Import from PeopleSoft dialog box to help you identify the tree whose attributes you want to import:

- **SetID.** If a tree has a SetID, it appears before the tree name.
- **Tree name.** The name of the tree.
- **Effective date.** The tree effective date appears after the tree name.

The Import Attributes from PeopleSoft dialog box shows only one folder in the Trees tab, where it lists all available trees.

Importing the Detail Record

When you import a detail tree, you import the tree metadata only. Similarly, when you import the attributes of a detail tree into a created tree source definition, you import the tree metadata only. To join the created tree with the detail record, you must import the detail record also.

Imported Detail Trees

The tree source definition of an imported tree displays the name of the associated detail record. The detail record name appears in the Detail Table Name field on the Properties tab when you edit the source definition. The column name for detail data appears in Detail Field Name.

Import the detail record. You can join the tree and detail record in a single Application Source Qualifier in a mapping.

Created Tree Source Definitions

A created tree source definition does not display the name of the associated detail record. To determine the associated detail record, refer to the PeopleSoft software.

Import the detail record. Use a Joiner transformation to join the created tree source definition and the detail record.

Connecting to the PeopleSoft System

To import PeopleSoft objects, use the Import from PeopleSoft dialog box to connect to the PeopleSoft system. You can enter a filter to reduce the number of objects the Designer displays in the selection list. You can also enter PeopleSoft language codes to specify the language in which you want language-sensitive metadata, such as column names and descriptions.

You can import a PeopleSoft objects from the Source Analyzer and Mapping Designer. When you import PeopleSoft objects from the Source Analyzer, you can import PeopleSoft records and PeopleSoft trees for horizontal flattening. When you import PeopleSoft objects from the Mapping Designer, you can import tree attributes into a created tree source definition.

Entering Connection Information

The Designer uses an ODBC data source and database username and password to connect to the PeopleSoft metadata tables to import PeopleSoft object definitions or tree attributes. The database username you enter must have SELECT permission on the PeopleSoft metadata tables.

If the database username is the owner of the PeopleSoft metadata tables, you can leave the Owner field blank. If the database username is *not* the owner of the PeopleSoft metadata tables, you must enter the owner name in the Import from PeopleSoft dialog box. This allows the Designer to access the metadata tables to import PeopleSoft source definitions.

Filtering Available Records and Trees

When you import PeopleSoft sources, you can enter a filter condition to limit the number of records and trees displayed in the Import from PeopleSoft dialog box. If you do not enter a filter, the Designer displays all available records and trees in the PeopleSoft system. You might use the filter to improve the speed at which the Designer displays records and trees.

You can filter records and trees as follows:

- Records on the Records tab based on the record or business name
- Trees on the Trees tab based on tree name

Note: When you import tree attributes for a created tree source definition in the Mapping Designer, the Import Attributes from PeopleSoft dialog box only displays the Trees tab.

You cannot filter records that appear on the Panels tab, since they are organized by the PeopleSoft organizational hierarchy.

When you enter a filter, use the following guidelines:

- Enter one filter criterion at a time.
- Enter case-sensitive text.
- Use an underscore (_) or the percent symbol (%) as wildcard characters. Use an underscore to represent a single character. Use the percent symbol to represent multiple characters.

For example, to find a PeopleSoft record with “AGING” as part of the record name, enter the following filter:

```
%AGING%
```

The filter returns records containing AGING in the record name. If you know the record name begins with AGING, you can enter the filter without the first wildcard.

To perform a similar filter on PeopleSoft business names, which can contain a mix of uppercase and lowercase text, enter `%Aging%` or `Aging%`.

To change a filter condition, enter the new condition and reconnect to the database.

Tip: When filtering trees, SetID and effective date are not part of the actual tree name.

Importing PeopleSoft Source Definitions

You can import PeopleSoft records and trees for horizontal flattening. When you import PeopleSoft sources, you add them to an import list. You can import more than one PeopleSoft object at once by making multiple selections and adding them to the import list. Once the import list is complete, you can import all listed PeopleSoft objects at once.

To import a PeopleSoft source definition:

1. In the Source Analyzer, click Sources > Import From PeopleSoft.

   After you connect to a PeopleSoft system, the Panels/Pages tab changes to the Panels tab or the Pages tab, depending on the version of PeopleSoft you are running. For example, if you connect to a PeopleSoft 8 system, the Panels/Pages tab changes to Panels. If you connect to an earlier version of PeopleSoft, the Panels/Pages tab changes to Pages.

2. Select the PeopleSoft ODBC connection, and then enter the database username and password to connect to the PeopleSoft metadata tables in the underlying database.

   Note: This username must have SELECT permission on the PeopleSoft metadata tables.

   If you need to create or modify an ODBC data source, click the button next to the list of existing data sources to open the ODBC Administrator. Create the appropriate data source, and click OK. In the Import Tables dialog box, select the new ODBC data source.

3. Optionally, enter the owner name of the PeopleSoft metadata tables.

   If the database username you entered is not the owner of the PeopleSoft metadata tables, enter the table owner name to view available records and trees.

4. Optionally, enter a filter criterion and how to apply the filter.

   When entering a filter criterion, use an underscore (_) as a wildcard to represent a single character. Use the percent symbol (%) to represent multiple characters.

   The Designer filters the records on the Records tab and trees on the Trees tab. The Designer does not filter records on the Panels tab.

   Tip: To filter table names, enter capitalized text. To filter business names, capitalize the initial letter of each word.

5. Optionally, enter a PeopleSoft language code.

   To import metadata in the base language of the PeopleSoft system, leave this option blank.

   If you want the Designer to import language-sensitive metadata in a particular language, enter the corresponding PeopleSoft language code.

6. Click Connect.

7. Select the Records, Panels, or Trees tab.

   PeopleSoft records appear on the Records tab and the Panels tab. Strict-level PeopleSoft trees appear on the Trees tab. Winter trees appear in the Winter trees folder. Detail and summary trees appear in the Trees folder.

8. Open the folders for the objects you want to import.

   Select the object or objects and click Add to Import List. You can add both records and trees to the import list.

9. Select the objects you want to add from a single tab and click Add to Import List.

   Select a different tab to add other objects. You can select and add multiple objects to the import list at one time:

   - Hold down the SHIFT key to select blocks within one folder.
   - Hold down the CTRL key to make non-contiguous selections within a folder.
   - Use the Select All button to select all tables within a folder.

   To clear all selected sources, click Select None. To enter a different filter, enter the new filter criterion and click Re-Connect.

10. To see the objects on the Import List, click View Import List.

    The Import List dialog box appears.

Tip: To remove an object from the list, select it and click Delete.

11. To close the Import List, click Close.
12. To import all listed PeopleSoft objects, click OK.

After you import a PeopleSoft source, the Designer displays it in the Navigator below the Sources node in the folder. The Designer organizes PeopleSoft sources by the ODBC data source used to import the source and the PeopleSoft object type. Records appear in a folder named *DataSourceName*_Records. Trees appear in a folder named *DataSourceName*_Trees. If the source record, panel, or tree changes after you create the source definition, reimport or update the source definition accordingly.

Creating Tree Source Definitions

To import a tree for vertical flattening, complete the following tasks:

1. Create a tree source definition in the Source Analyzer.
2. Include the tree source definition in a mapping and import the PeopleSoft tree attributes in the source definition instance.

You can use a tree source definition multiple times in a mapping and import the attributes of different PeopleSoft trees into each instance of the source definition.

Step 1. Create a Tree Source Definition

Create one tree source definition for all mappings.

To manually create a tree source definition:

1. In the Source Analyzer, click Sources > Create.

   The Create Source dialog box appears.
2. Enter a name for the tree source definition in the appropriate field.
3. Select PeopleSoft as the database type.
4. Enter a name in the Database Name field.

   In the Navigator, the database name appears below the Sources node in the folder. All created tree source definitions defined with this database name appear below it in the Navigator.
5. Click Create. Click Done.

The source definition does not contain any tree attributes. You import tree attributes for each instance of the source definition you include in a mapping.

Step 2. Import Tree Attributes

You can add multiple instances of the source definition to a mapping. For each instance, import the attributes of a PeopleSoft tree. The Designer uses the tree attributes to associate a PeopleSoft tree with the instance of the definition.

To import tree attributes into a tree source definition:

1. In the Mapping Designer, create a mapping.
2. Drag a created tree source definition into the workspace.
3. Double-click the source definition and click the Properties tab.

The attributes are empty.

4. Click the Add button in the Import Vertical Tree Attributes field.

   The Import Attributes from PeopleSoft dialog box appears.

5. Select the PeopleSoft ODBC connection.

   To create or modify an ODBC data source, click the button next to the list of existing data sources to open the ODBC Administrator. Create the appropriate data source, and click OK. Then in the Import Tables dialog box, select the new ODBC data source.

6. Enter the database username and password to connect to the PeopleSoft metadata tables in the underlying database.

   Note: The username must have SELECT permission on the PeopleSoft metadata tables.

7. Optionally, enter the owner name of the PeopleSoft metadata tables.

   If the database username is not the owner of the PeopleSoft metadata tables, enter the table owner name to view available records and trees.

8. Optionally, enter a filter criterion and how to apply the filter.

   You can filter tree names using the Table option, but not business names. The Designer returns all trees if you apply the filter to business names.

   Use an underscore (_) as a wildcard to represent a single character. Use the percent symbol (%) to represent multiple characters.

   The Designer filters trees on the Trees tab. To filter tree names, enter capitalized text.

9. Optionally, enter a PeopleSoft language code.

   To import metadata in the base language of the PeopleSoft system, leave this option blank. If you want the Designer to import language-sensitive metadata in a particular language, enter the corresponding PeopleSoft language code.

10. Click Connect.

    Strict-level and loose-level trees appear on the Trees tab below the Trees folder.

11. Select the tree and click OK.

    The Designer displays the tree attributes on the Properties tab. If there are attributes you do not want, click Revert to remove them.

    The Import Vertical Tree Attributes field is empty after you import the tree attributes.

12. Click OK to close the Edit Transformations dialog box.

If the tree changes after you import the tree attributes, reimport or update the tree attributes accordingly.

Editing PeopleSoft Source Definitions

You can edit a PeopleSoft source definition to add descriptions or accommodate minor changes to the source. You can change source or port names, enter descriptions, change the port order, or delete ports. You can also change key types for PeopleSoft records to create primary-foreign key relationships.

Note: Because source definitions must match the source, import definitions instead of creating them manually.

To edit a PeopleSoft source definition:

1. In the Source Analyzer, double-click the title bar of the source definition.

2. Edit the following settings:

Table Settings	Description
Select Table	Displays the source definition you are editing. To choose a different open source definition to edit, select it from the menu.
Rename button	Opens a dialog box to edit the following properties: - Source name - Database name - Business name
Owner Name	Owner of the business component or table.
Description	Description of the source definition.

3. Click the Columns tab.
4. From the Columns tab, you can edit column names, datatypes, key types, and other properties.
5. Click the Metadata Extensions tab.
6. From the Metadata Extensions tab, you can add, remove, or edit metadata extension names, datatypes, key types, and other restrictions.
7. Click OK.

CHAPTER 4

Application Source Qualifier for PeopleSoft Sources

This chapter includes the following topics:

- Application Source Qualifier for PeopleSoft Sources Overview, 36
- Understanding the Default Query, 38
- Joining Source Data, 39
- Understanding the Default Join, 40
- Entering a Source Filter, 44
- Entering a Join Override, 46
- Using an Extract Override, 47
- Sorting Ports, 49
- Selecting Distinct Values, 49
- Selecting Current Rows, 50
- Linking the TO_EFFDT Port, 50
- Joining Effective Dated PeopleSoft Records, 51
- Configuring an Application Source Qualifier, 58

Application Source Qualifier for PeopleSoft Sources Overview

When you use a PeopleSoft source definition in a mapping, you connect it to an Application Source Qualifier. The Application Source Qualifier represents the record set you want the PowerCenter Integration Service to return when you run the session. Use the Application Source Qualifier to represent application data in PeopleSoft systems.

The Designer generates a query to extract source data based on the Application Source Qualifier connected ports, transformation properties, and the relationships between connected and associated source definitions.

In the Application Source Qualifier, you can complete the following tasks:

- **Specify sorted ports.** When you specify a number for sorted ports, the Designer adds an ORDER BY clause to the default query for PeopleSoft records. This orders source data by the number of ports you enter.

- **Select only distinct values from the source.** When you choose Select Distinct, the Designer adds a SELECT DISTINCT statement to the default query. This returns only distinct source rows.
- **Join data from the same PeopleSoft system.** When you join sources, the Designer adds the join conditions to a WHERE clause in the default query. You can join two or more PeopleSoft records with key relationships by connecting them to one Application Source Qualifier. You can also join a detail tree with its detail record for imported tree sources. Use the default join or create a custom join.
- **Filter data.** When you enter a source filter, the Designer adds the filter conditions to a WHERE clause in the default query. Use a source filter to remove source rows you do not need in the data flow.
- **Create a custom extract query.** You can create an extract override for PeopleSoft records. An extract override replaces the default query. You can create an extract override that includes defined transformation settings, or you can override all existing transformation settings.
- **Extract current data.** You can include a WHERE clause in the default query to return only the current rows according to the date you specify.

The Application Source Qualifier includes the following transformation options:

- Filter
- Join Override
- Extract Override
- Number of Sorted Fields
- Tracing Level
- Select Distinct
- Extract Current Rows
- Extract Date
- Effective Date Join Order

The Properties tab of the Application Source Qualifier only shows the applicable transformation options for the particular source definition. For example, the Extract Current Rows option only appears for Application Source Qualifiers with effective dated PeopleSoft records.

Using Table Names for PeopleSoft Records

PeopleSoft saves data for a PeopleSoft record in an underlying database table. By default, the database table name is PS_*Record_Name*. Since the PowerCenter Integration Service extracts data from underlying database tables, the Designer uses the database table name in the default query.

Make sure you use the database table name rather than the source definition name when you alter the default query using any of the following transformation options:

- Filter
- Join Override
- Extract Override

  Tip: To ensure you have the correct database table name, select ports from the Ports column in the Source Editor instead of manually entering the table name.

Using Parameters and Variables with PeopleSoft Sources

Use mapping parameters and variables in the following transformation properties of an Application Source Qualifier transformation for PeopleSoft. You can also use the system variable $$$SessStartTime:

- Extract Override
- Join Override
- Filter
- Extract Date

Note: When you use a mapping parameter or variable for the Extract Date transformation option, the Designer includes single quotes around the mapping parameter or variable when you generate the SQL statement in the Extract Override option. Verify that the SQL statement in the extract override complies with the underlying database.

You can also use mapping parameters and variables in the following transformation properties in the source definition of a *created* tree source definition in the Mapping Designer:

- Tree Name
- SetID
- Effective Date
- Set Control Value

Understanding the Default Query

The default query is a SELECT statement in the Application Source Qualifier transformation. The Designer creates the default query based on the following:

- Connected transformation input and output ports
- Transformation settings
- Relationships between sources connected to the same Application Source Qualifier

The PowerCenter Integration Service uses the default query as the basis for the extract query for the session. The following factors cause the PowerCenter Integration Service to alter or expand the default query:

- The PowerCenter Integration Service process code page differs from the code page of the PeopleSoft underlying database, which requires the PowerCenter Integration Service to perform code page translations.
- You enter a PeopleSoft language code in the application connection for a PeopleSoft source, which requires the PowerCenter Integration Service to join source tables with their related language tables.

The PowerCenter Integration Service writes the extract query into the session log.

The default query includes only the ports in the Application Source Qualifier that are connected to other transformations.

Although all ports in the source definition are connected to the Application Source Qualifier, only three ports are connected to the Expression transformation. In this case, the Designer generates a default query that selects only those three columns:

```
SELECT PS_PROJ_VARY_BA_VW.SETID, PS_PROJ_VARY_BA_VW.BUSINESS_UNIT,
PS_PROJ_VARY_BA_VW.PROJECT_ID FROM PS_PROJ_VARY_BA_VW
```

Editing the Default Query

You can edit the default query in the Application Source Qualifier by editing the transformation properties. This includes editing the transformation settings or changing the connected ports in the transformation.

When you edit the Application Source Qualifier, the Designer includes these settings in the default query. The PowerCenter Integration Service uses the default query to create the extract query for the session.

You can also override the default query with the Extract Override option. When you enter an Extract Override, the PowerCenter Integration Service uses the defined extract override.

Viewing the Default Query

You can view the default query for PeopleSoft records in the Application Source Qualifier transformation. For accurate results, connect all the input and output ports you want to use in the mapping before generating the query.

To view the default query for PeopleSoft records:

1. From the Properties tab, select Extract Override.

   The SQL Editor appears.

   Note: If a source definition for a PeopleSoft tree is connected to or associated with the Application Source Qualifier transformation, the Extract Override option is disabled.
2. Select Generate SQL.

   Unless you change or override this query, the PowerCenter Integration Service uses this query to generate the extract query.
3. Click Cancel to exit.

   If you do not cancel the extract override, the PowerCenter Integration Service uses the resulting query. It ignores subsequent changes you make to the transformation and does not perform automatic language code translations.

Joining Source Data

If you have multiple PeopleSoft sources in a mapping, you can connect them to a single Application Source Qualifier. This allows the PowerCenter Integration Service to read all connected tables in a single pass. Single pass reads can reduce the total connection time to the source system and improve session performance.

You can connect multiple PeopleSoft sources to an Application Source Qualifier if all of the following conditions are true:

- All sources are located in the same database.
- All sources have logical relationships.
- All sources are imported, not created in the Designer.

When you join sources in one Application Source Qualifier, the Designer creates a join based on the relationships between sources.

Note: If you need to join a created tree source definition with another source, for example a detail record, you must use a Joiner transformation.

The Application Source Qualifier provides two types of joins:

- **Default join.** The Designer creates a default join based on connected import ports and the implicit relationships between connected or associated sources.
- **Join override.** You can enter a join override to customize the default join.

Understanding the Default Join

When you connect multiple PeopleSoft source definitions to one Application Source Qualifier, the Designer joins them based on logical relationships between connected and associated sources.

The following table describes how the Designer can create a default join for PeopleSoft source definitions:

Source Type	Join With	Default Join Condition
PeopleSoft record	PeopleSoft record	Joins if records are related. Records are related when both of the following are true: - Records contain source columns with matching names. - At least one of the matching-name columns is a PeopleSoft key. Records are also related if you define a primary-foreign key relationship in the Source Analyzer. If records are not related, you can join them with a join override.
Imported PeopleSoft detail tree	Detail record	None. You can always join an imported detail tree with the detail record.
Imported PeopleSoft detail tree	Non-detail record	Joins if all of the following are true: - The record is related to the detail record for the imported tree. - The detail record is connected to or associated with the Application Source Qualifier.

Joining Records

When you connect multiple PeopleSoft record source definitions to one Application Source Qualifier, the Designer creates the default join based on logical key column and column name relationships. The Designer can create a default join of PeopleSoft records when both of the following conditions are true:

- Source columns have matching names.
- At least one matching-name column is a PeopleSoft key.

The Designer also creates a default join of PeopleSoft records if you create primary-foreign key relationships in the Source Analyzer.

You can, therefore, join two records in the following cases:

- **Sources have PeopleSoft key columns with matching names.** The Designer joins two key columns if they have matching names. Columns must be PeopleSoft keys.

  For example, you connect two PeopleSoft sources to an Application Source Qualifier. Both sources have a PeopleSoft key column named EmployeeNumber. The Designer joins the two columns.
- **A PeopleSoft key column name matches a non-key column name.** The Designer joins two columns when a PeopleSoft key column name in one source matches a non-key column name in another source.

For example, if one of the connected sources has a PeopleSoft key column named EmployeeNumber and another source has a column of the same name marked not-a-key, the Designer joins the two columns.

- **Sources have a user-defined primary-foreign key relationship.** The Designer joins two key columns if they have a primary-foreign key relationship.

To join sources, you connect source columns to one Application Source Qualifier. The default join is an inner equijoin. The PowerCenter Integration Service generates a SELECT statement that includes columns used for the join in the WHERE clause:

```
Source1.Column_Name = Source2.Column_Name AND
Source1.Column_Name = Source2.Column_Name...
```

If the default join does not join source tables the way you want, you can enter a join override in the Application Source Qualifier.

By default, the PowerCenter Integration Service joins all rows from all PeopleSoft records. When you connect one or more effective dated records to an Application Source Qualifier, you can alter the default join by selecting certain transformation options.

Related Topics:

- "Joining Effective Dated PeopleSoft Records" on page 51

Joining Detail Trees and Detail Records

You can join detail trees with their detail records. However, joining imported trees with records is different than joining created tree source definitions with records.

You can join a detail tree that you import with its detail record in the Application Source Qualifier. The detail record contains values related to the detail ranges in a tree.

Note: To join a created tree source definition with any other source, use a Joiner transformation. You can only associate one created source definition with the Application Source Qualifier.

When you connect an imported detail tree and detail record in an Application Source Qualifier, the Designer creates a default join based on the detail ranges in the tree and the related column in the detail record. When the PowerCenter Integration Service performs the join, it produces a row for each value in the related column of detail record that matches or falls within the tree detail ranges. When the PeopleSoft tree references a detail value more than once, the PowerCenter Integration Service produces a row for each use of the detail value.

For example, the following data represents the detail tree pictured in "Detail Trees and the Detail Record" on page 17.

The following table displays the RANGE_FROM and RANGE_TO values for each row that represent the detail ranges in the tree:

Country	Region	State	RANGE_FROM	RANGE_TO
US	E	NY	1453	1453
US	E	DC	1480	1480
US	W	NULL	534	534

Country	Region	State	RANGE_FROM	RANGE_TO
US	W	CA	2334	2434
US	W	CA	3001	3001

The following table displays the detail record for the tree, which contains the related column, Store_ID, and other columns including Mgr_ID and Employee:

Store_ID	Mgr_ID	Employee
534	5401	15
1453	7001	43
1480	1101	23
2334	1601	63
2432	301	52
3001	2001	19
3234	1401	32

The following table shows the results of the default join when you connect the detail tree and detail record in the Application Source Qualifier:

Country	Region	State	RANGE_FROM	RANGE_TO	Store_ID	Mgr_ID	Employee
US	E	NY	1453	1453	1453	7001	43
US	E	DC	1480	1480	1480	1101	23
US	W	NULL	534	534	534	5401	15
US	W	CA	2334	2434	2334	1601	63
US	W	CA	2334	2434	2432	301	52
US	W	CA	3001	3001	3001	2001	19

The US-W-CA-2334-2434 row from the tree appears twice, once for each value in the Store_ID column that falls within the range, 2334 to 2434. The 3234 store ID row from the detail record does not appear in the result set because it is not included in the detail ranges of the tree. The PowerCenter Integration Service replaces the missing node in the US-W-534 row with NULL.

Joining Detail Trees with Non-Detail Records

You can join the source definition of a detail tree that you import and a non-detail record source definition in an Application Source Qualifier transformation. The Designer can perform this default join if both of the following conditions are true:

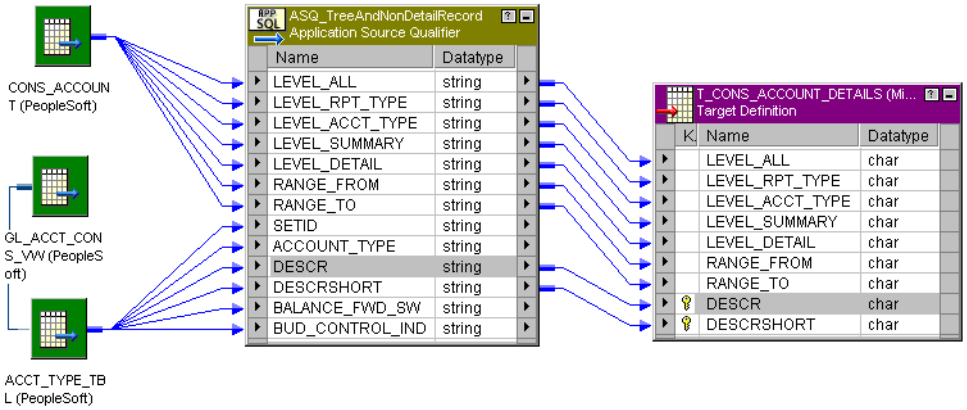
- The non-detail record is related to the detail record for the tree.
- The detail record for the tree is connected or associated with the Application Source Qualifier.

Note: To join a created tree source definition with any other source, use a Joiner transformation. You can only associate one created source definition with the Application Source Qualifier.

When you join an imported detail tree source definition with a non-detail record source definition related to the detail record, the Designer uses the relationship between the records to create the default join.

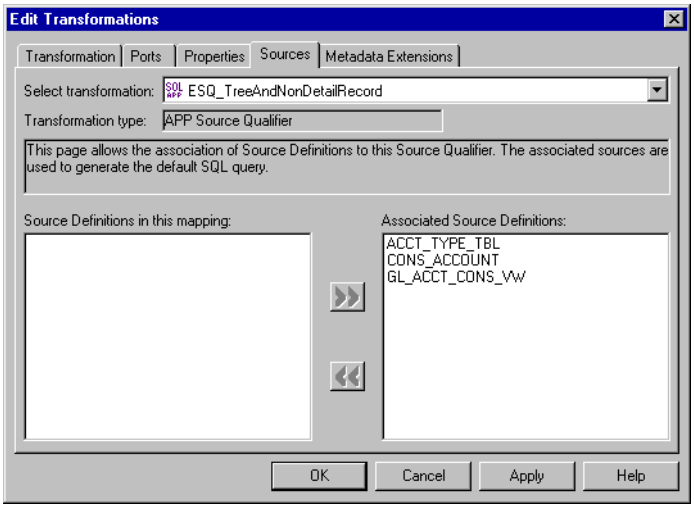
When designing a mapping, you can connect a detail tree, detail record, and non-detail record to the same Application Source Qualifier. However, if you do not require data from the detail record, you can leave the detail record unconnected. The Designer validates the mapping as long as the detail record is associated with the Application Source Qualifier performing the join.

For example, the mapping in the following figure joins the tree CONS_ACCT and a non-detail record ACCT_TYPE_TBL:



The mapping contains the detail record GL_ACCT_CONS_VW. The Designer validates the default join of the tree and non-detail record sources because the detail record is an associated source, even though the detail record is not connected to the Application Source Qualifier.

The following figure shows the GL_ACCT_CONS_VW detail record associated with the Application Source Qualifier transformation:



Entering a Source Filter

You can enter a filter condition to reduce the number of source rows the PowerCenter Integration Service returns from PeopleSoft sources. You can enter a single filter condition or a series of conditions. For example, you might create a filter if you want the PowerCenter Integration Service to return only data falling within a range of dates.

If you have one or more effective dated PeopleSoft records connected to the Application Source Qualifier, you can filter source rows based on current dates. Use the Extract Current Rows transformation option to alter the default query to select only the current rows.

Note: When you enter a source filter on the Partitions tab of the session properties, you override the source filter in the Application Source Qualifier and on the Transformations tab of the session properties.

Filter Syntax

You can enter a filter condition using any syntax supported by the underlying database for the PeopleSoft system. When you enter a filter condition, the Designer adds a WHERE clause to the default query. You enter only the condition following the WHERE. You can enter multiple filter conditions by using AND or OR. You can define a filter using any port except the TO_EFFDT port of an effective dated PeopleSoft record.

For example, enter the following filter condition if you want the database to return all SYSDATE dates in March, April, and May 2001:

```
Source1.Sysdate >= '3/1/2001' AND Source1.Sysdate =< '5/31/2001'
```

Note: If you define language codes in PeopleSoft source database connections, do not include language-sensitive columns in filter conditions. Otherwise, the PowerCenter Integration Service might return incomplete or inaccurate data.

Creating a Source Filter

Use the following procedure to create a source filter.

To create a source filter:

1. In the Mapping Developer, edit an Application Source Qualifier transformation, and select the Properties tab.
2. Click the right corner of the Filter option to open the Source Editor.
3. Enter a filter condition supported by your source database.

   Use SQL syntax supported by the underlying database for the PeopleSoft source system. Do not enter WHERE.

   Use database tables names in the filter syntax. Database table names for PeopleSoft record sources typically begin with PS_. For example, the database table name for the PeopleSoft record PROJ_MILE_TBL is PS_PROJ_MILE_TBL.

   Select port names from the Ports tab to ensure the database table name is correct. You can define a filter using any port except the TO_EFFDT port of an effective dated PeopleSoft record.

   Important: Do not use language-sensitive columns in the filter.
4. Click OK to return to the Edit Transformations dialog box. Click OK again to return to the Designer.

Validating Filter Syntax

You can validate filter syntax for PeopleSoft records with the Validate button in the Extract Override option. When validating filter syntax, note that any errors that appear might be related to other options configured in the Application Source Qualifier.

To validate filter syntax, you need a source database username and password and the ODBC data source used to connect to the source.

To validate filter syntax:

1. In the Mapping Developer, edit the Application Source Qualifier, and select the Properties tab.
2. Click the right corner of the Extract Override option.

   If a source definition for a PeopleSoft tree is connected to or associated with the Application Source Qualifier transformation, the Extract Override option is disabled.
3. Click Generate SQL.

   The Designer generates the extract SQL query using the filter condition and any other conditions entered in the transformation.
4. Select the ODBC Data Source used to connect to the source system, and enter the database username and password to connect to the database.
5. Click Validate.

   The Designer connects to the source database and tests the query syntax. The Designer displays a message that states whether the syntax is valid.
6. Click Cancel to cancel the Extract Override.

   If you do not cancel the Extract Override, the Designer uses the generated SQL as an extract override. The PowerCenter Integration Service uses the extract override. It ignores subsequent changes you make to the transformation and does not perform automatic language code translations.
7. Click Cancel to close the transformation or correct and validate the filter.

Entering a Join Override

You can create a custom join to override the default join in the Application Source Qualifier. In the join, you can enter any expression supported by the underlying database for the PeopleSoft source system. The Designer appends the join to existing WHERE clauses in the default query.

You might create a join override under the following conditions:

- **The names of key columns do not match.** For example, you might have one source with a key column in Source1 named EmployeeNum. You want to join the source with another with a non-key column in Source2 named EmployeeNumber. Since the column names differ, the Application Source Qualifier does not join them. You can enter the following join override:

  ```
  Source1.EmployeeNum = Source2.EmployeeNumber
  ```

- **To perform an outer join**. For example, you enter a left outer join to pass all of Source1 and join Source2 with Source1 where the Social_Security columns are equal:

  ```
  Source1.Social_Security(+) = Source2.Social_Security
  ```

You can define a join override using any column except the TO_EFFDT column of an effective dated PeopleSoft record.

To create a join override:

1. Create an Application Source Qualifier transformation containing data from multiple sources or associated sources.
2. Edit the Application Source Qualifier and select the Properties tab.
3. Click the right corner of the Join Override option to open the Source Editor.
4. Enter the syntax for the join.

   Enter only statements supported by the underlying database for the PeopleSoft source system.

   Use database tables names in the join syntax. Database table names for PeopleSoft record sources typically begin with PS_. For example, the database table name for the PeopleSoft record PROJ_MILE_TBL is PS_PROJ_MILE_TBL.

   Select port names from the Ports tab to ensure the database table name is correct. You can define a join override using any port except the TO_EFFDT port of an effective dated PeopleSoft record.
5. Click OK to return to the Edit Transformations dialog box. Click OK again to return to the Designer.

The Application Source Qualifier adds the join override to the default query.

When you create a custom join, the Designer assumes the join is valid. If the join is not valid, the session fails. You can validate a join override.

Validating Join Override Syntax

You can validate the syntax of a join override for PeopleSoft records with the Validate button in the Extract Override option. When validating the join override syntax, any errors that appear might be related to other options configured in the Application Source Qualifier.

To validate join override syntax, you need a source database username and password and the ODBC data source used to connect to the source.

To validate join override syntax:

1. In the Mapping Developer, edit the Application Source Qualifier in which you have a join override, and click the Properties tab.
2. Click the right corner of the Extract Override option.

If a source definition for a PeopleSoft tree is connected to or associated with the Application Source Qualifier transformation, the Extract Override option is disabled.

3. Click Generate SQL.

   The Designer generates the extract SQL query using the override join condition and any other conditions entered in the transformation.

4. Select the ODBC Data Source used to connect to the source system, and enter the database username and password to connect to the database.

5. Click Validate.

   The Designer connects to the source database and tests the query. The Designer displays a message stating whether the syntax of the query is valid.

6. Click Cancel to cancel the Extract Override.

   If you do not cancel the Extract Override, the Designer uses the generated SQL as an extract override. The PowerCenter Integration Service uses the extract override. It ignores subsequent changes you make to the transformation and does not perform automatic language code translations.

7. Click Cancel to close the transformation or correct and validate the join override.

Using an Extract Override

You can create an extract override to override the Application Source Qualifier default query for PeopleSoft records. Use any SQL statement supported by the underlying database for the PeopleSoft source system. The PowerCenter Integration Service uses the extract override to extract data from connected sources.

Use an extract override for an Application Source Qualifier connected to PeopleSoft records only. The PowerCenter Integration Service does not perform extract overrides for Application Source Qualifiers connected to PeopleSoft trees. When you connect an Application Source Qualifier to a PeopleSoft tree, the Designer disables the Extract Override option.

When you create an extract override for PeopleSoft records, you can either:

- **Generate and edit the default query.** To use the existing transformation options in the extract override, generate and edit the default query. When the Designer generates the default query, it incorporates all other configured options, such as a filter or number of sorted ports. The resulting query overrides all other options you might subsequently configure in the transformation.
- **Manually enter the entire extract override.** The resulting query overrides all other options configured in the transformation.

For example, an Application Source Qualifier is already configured with the following join override:

```
PS_AB_CREDITBU_VW.CUST_ID = PS_AB_CREDITCP_VW.CORPORATE_CUST_ID AND
PS_AB_CREDITBU_VW.CR_LIMIT = PS_AB_CREDITCP_VW.CR_LIMIT AND
PS_AB_CREDITCP_VW.CR_LIMIT_RANGE > 100
```

When you generate the default query, the Designer incorporates the join override in the WHERE clause:

```
SELECT PS_AB_CREDITCP_VW.CORPORATE_SETID, PS_AB_CREDITCP_VW.CORPORATE_CUST_ID,
PS_AB_CREDITCP_VW.NAME1, PS_AB_CREDITCP_VW.ROLEUSER, PS_AB_CREDITCP_VW.BAL_AMT,
PS_AB_CREDITCP_VW.CR_LIMIT, PS_AB_CREDITCP_VW.CR_LIMIT_CORP,
PS_AB_CREDITCP_VW.CR_LIMIT_REV_DT, PS_AB_CREDITCP_VW.CR_LIMIT_RANGE,
PS_AB_CREDITCP_VW.CURRENCY_CD, PS_AB_CREDITBU_VW.SETID, PS_AB_CREDITBU_VW.CUST_ID,
PS_AB_CREDITBU_VW.NAME1, PS_AB_CREDITBU_VW.ROLEUSER, PS_AB_CREDITBU_VW.BAL_AMT,
PS_AB_CREDITBU_VW.CR_LIMIT, PS_AB_CREDITBU_VW.CR_LIMIT_REV_DT,
PS_AB_CREDITBU_VW.CR_LIMIT_RANGE, PS_AB_CREDITBU_VW.CREDIT_CLASS,
PS_AB_CREDITBU_VW.CURRENCY_CD
FROM PS_AB_CREDITCP_VW, PS_AB_CREDITBU_VW
```

```
WHERE
PS_AB_CREDITBU_VW.CUST_ID = PS_AB_CREDITCP_VW.CORPORATE_CUST_ID AND
PS_AB_CREDITBU_VW.CR_LIMIT = PS_AB_CREDITCP_VW.CR_LIMIT AND
PS_AB_CREDITCP_VW.CR_LIMIT_RANGE > 100
```

To use the join override in the extract override, you can generate the default query, and then edit the rest of the query, rather than entering the entire extract override.

Verify that the SQL statement syntax in the extract override is valid for the underlying database of your PeopleSoft system. For example, when you generate and edit the default query for an effective dated PeopleSoft record, you might need to edit the SQL statement. If you link the TO_EFFDT port to another transformation or target, the default query includes a NULL statement. If the underlying database is Oracle, for example, you need to change NULL to to_date(NULL).

Note: When you enter an extract override on the Partitions tab of the session properties, you override the extract override in the Application Source Qualifier and on the Transformations tab of the session properties.

Creating an Extract Override

To incorporate configured transformation options in the Application Source Qualifier in an extract override, generate and customize the default query. Before generating the default query, connect the Application Source Qualifier transformation to all subsequent transformations. The Designer uses both connected input and output ports to generate the appropriate extract query.

Once you enter an extract override, the Designer accepts the override and ignores any subsequent changes you might make to the Application Source Qualifier transformation. The PowerCenter Integration Service uses the extract override to query sources.

Note: When you enter an extract override, the PowerCenter Integration Service does not perform language code queries. It ignores any language code entered in the session database connection.

To create and validate an extract override:

1. In the Mapping Developer, edit an Application Source Qualifier transformation, and select the Properties tab.
2. Click the right corner of the Extract Override option to open the Source Editor.
3. To use existing transformation properties in the override, click Generate SQL, and then edit the extract query.

   To ignore any existing transformation configurations, enter the extract override. Use SQL syntax supported by the underlying database for the PeopleSoft source system.

   Use database table names in the extract override. Database table names for PeopleSoft record sources typically begin with PS_. For example, the database table name for the PeopleSoft record PROJ_MILE_TBL is PS_PROJ_MILE_TBL.

   Select port names from the Ports tab to ensure the database table name is correct.
4. To validate extract override syntax, select the ODBC data source used to connect to the source database, enter the database username and password and click Validate.

   The Designer connects to the source database and tests the query syntax. The Designer displays a message stating whether the syntax is valid.
5. Click OK to save the query and return to the Edit Transformations dialog box.
6. Click OK again to return to the Designer.

The Designer replaces the default query with the extract override. The PowerCenter Integration Service uses this override to query sources unless you update or delete the extract override. The PowerCenter Integration Service does not append language code queries to an extract override.

Sorting Ports

You can sort source data from PeopleSoft records using the Number of Sorted Ports option in the Application Source Qualifier. When you specify the Number of Sorted Ports, the Designer adds an ORDER BY clause to the default query and then includes the configured number of PeopleSoft record ports in the ORDER BY clause. The Designer adds ports to the ORDER BY clause starting with the ports appearing at the top of the Application Source Qualifier.

For example, you set the number of sorted ports to 2 in an Application Source Qualifier. The Designer adds an ORDER BY for the default query that includes the first two ports in the transformation, as follows:

```
SELECT PS_PROJ_VARY_BA_VW.SETID, PS_PROJ_VARY_BA_VW.BUSINESS_UNIT,
PS_PROJ_VARY_BA_VW.PROJECT_ID FROM PS_PROJ_VARY_BA_VW
ORDER BY PS_PROJ_VARY_BA_VW.SETID, PS_PROJ_VARY_BA_VW.BUSINESS_UNIT
```

You might use sorted ports to improve performance in Aggregator transformations or to ensure the PowerCenter Integration Service reads columns in a specified order.

The Number of Sorted Ports option applies to ports in a PeopleSoft record. When the Application Source Qualifier is connected only to a PeopleSoft tree, it ignores the Number of Sorted Ports option. When the Application Source Qualifier is connected to a tree and records, it sorts only ports connected to records, ignoring ports connected to a tree.

You can also use a Sorter transformation to sort ports from a PeopleSoft source.

To enter a number of sorted ports:

1. In the Mapping Developer, edit an Application Source Qualifier transformation, and select the Properties tab.
2. Enter the number of ports you want the PowerCenter Integration Service to sort by.
3. Click the Ports tab.
4. Use the Up/Down arrows to move the ports you want the PowerCenter Integration Service to sort to the top of the list of ports.

   The Designer adds the configured number of ports to an ORDER BY clause in the default query, starting with the first listed port. The Designer adds only ports from PeopleSoft record sources.

Selecting Distinct Values

If you want the PowerCenter Integration Service to select only unique values from a PeopleSoft Record, use the Select Distinct option.

For example, you enable the select distinct in an Application Source Qualifier. The Designer adds SELECT DISTINCT to the default query, as follows:

```
SELECT DISTINCT PS_PROJ_VARY_BA_VW.SETID, PS_PROJ_VARY_BA_VW.BUSINESS_UNIT,
PS_PROJ_VARY_BA_VW.PROJECT_ID FROM PS_PROJ_VARY_BA_VW
```

You might use Select Distinct to extract unique customer IDs from a table listing total sales. You can filter out unnecessary data early in the data flow, which improves session performance.

By default, the PowerCenter Integration Service uses a SELECT statement to select all data.

To use Select Distinct:

1. In the Mapping Developer, edit an Application Source Qualifier transformation, and select the Properties tab.

2. Check Select Distinct.
3. Click OK to close the dialog box and save your changes.

   The Designer adds SELECT DISTINCT to the default query.

Selecting Current Rows

By default, the PowerCenter Integration Service extracts all rows from PeopleSoft records. However, you can specify to select only the current rows from the effective dated PeopleSoft record. Use the following transformation options in the Application Source Qualifier to alter the default query for effective dated PeopleSoft Records:

- Extract Current Rows
- Extract Date

When you select Extract Current Rows, the default query includes a WHERE clause to select rows from records that meet the current date requirements. Use the Extract Date option to specify the current date when extracting current rows. Leave the option blank to use the PowerCenter Integration Service current date.

Linking the TO_EFFDT Port

The Designer adds a new column, TO_EFFDT, to the source definition when you import an effective dated PeopleSoft record. The Designer determines the value of TO_EFFDT by the next value of EFFDT for the same key.

When you link TO_EFFDT to another transformation or target, the default query includes a self join to determine the value of TO_EFFDT. When there are no future rows for the same key, the Designer assigns NULL to TO_EFFDT.

The TO_EFFDT port in the Application Source Qualifier is linked to a port in another transformation and ultimately to the target. The following SQL statement is the default query for this Application Source Qualifier:

```
SELECT PS_EMP_SAL.EFFDT, PS_EMP_SAL.EMP_ID, PS_EMP_SAL.EMP_SAL, A0.EFFDT FROM
PS_EMP_SAL, PS_EMP_SAL A0 WHERE PS_EMP_SAL.EMP_ID = A0.EMP_ID AND A0.EFFDT IN (SELECT
MIN(EFFDT) FROM PS_EMP_SAL A1 WHERE A1.EMP_ID = PS_EMP_SAL.EMP_ID AND A1.EFFDT >
PS_EMP_SAL.EFFDT)UNION ALL SELECT PS_EMP_SAL.EFFDT, PS_EMP_SAL.EMP_ID,
PS_EMP_SAL.EMP_SAL, NULL
FROM PS_EMP_SAL WHERE PS_EMP_SAL.EFFDT IN (SELECT MAX(EFFDT) FROM PS_EMP_SAL A0
WHERE PS_EMP_SAL.EMP_ID = A0.EMP_ID)
```

The SQL statement above includes a self join. It selects a value for the TO_EFFDT port by selecting a different EFFDT value from the same table. The WHERE clause determines the particular EFFDT value it selects. If the PowerCenter Integration Service finds a row with the same key value with a greater value in EFFDT, it selects the greater EFFDT value as the value for TO_EFFDT.

The SQL statement includes a UNION ALL statement. After the UNION ALL statement, the query selects NULL for the TO_EFFDT port. It selects NULL for rows where the EFFDT value is the greatest in the table for the same key.

The following table shows an example of the data in the EMP_SAL table:

EFFDT	EMP_ID	SALARY
1990-01-01 00:00:00.000	1000	35,000.00
1992-01-01 00:00:00.000	1000	45,000.00
1994-01-01 00:00:00.000	1000	70,000.00
1990-01-01 00:00:00.000	9999	50,000.00

The following table displays the data when the PowerCenter Integration Service extracts data from this table and creates values for the TO_EFFDT port:

EFFDT	EMP_ID	SALARY	TO_EFFDT
1990-01-01 00:00:00.000	1000	35,000.00	1992-01-01 00:00:00.000
1992-01-01 00:00:00.000	1000	45,000.00	1994-01-01 00:00:00.000
1994-01-01 00:00:00.000	1000	70,000.00	NULL
1990-01-01 00:00:00.000	9999	50,000.00	NULL

Joining Effective Dated PeopleSoft Records

You can join multiple effective dated PeopleSoft records in an Application Source Qualifier transformation. The PowerCenter Integration Service joins effective dated PeopleSoft records in different ways, depending on the transformation options you choose.

The Application Source Qualifier includes three transformation options for effective dated PeopleSoft records. These options only appear if you connect one or more effective dated records to the Application Source Qualifier.

The following table lists the transformation options in the Application Source Qualifier for effective dated PeopleSoft records:

Application Source Qualifier Option	Description
Extract Current Rows	Extracts the current rows from effective dated records. Uses the date in Extract Date as the current date. When you use this option, the Designer modifies the default query and default join to select only the most recent data as defined by the extract date.
Extract Date	Defines the current date to use with the Extract Current Rows option. Uses the current date when this attribute is empty. Enter the date in the following format: `MM/DD/YYYY HH24:MI:SS`.
Effective Date Join Order	Specifies the join order between effective dated records. This option only appears when you connect multiple effective dated records to the Application Source Qualifier.

By default the PowerCenter Integration Service joins all rows from all records for each key in an effective dated record. However, you can join certain rows based on their effective dates and effective sequences by defining some of the options listed above.

When you specify to extract current rows, the PowerCenter Integration Service returns zero or one row for each key. When you connect only one record to the Application Source Qualifier, the PowerCenter Integration Service extracts the current row for each key in that record. When you connect multiple records to the Application Source Qualifier, you get different results. The results vary according to what you specify in the Effective Date Join Order option.

Use the Effective Date Join Order option to specify a join order for effective dated PeopleSoft records. The join order you specify determines which rows from each record the PowerCenter Integration Service joins. When you enter a join order, the PowerCenter Integration Service joins the row from the second table with a maximum effective date that is *less than or equal to* the effective date of the row from the first table.

You can specify a join order in the Application Source Qualifier in the following ways:

- Manually enter the PeopleSoft record names in the Effective Date Join Order field. Separate each record name with a comma.
- Click the arrow in the Effective Date Join Order field to open the Effective Date Join Order dialog box. Use the arrows in the dialog box to move the record names into the order and click OK.

The following table describes how the PowerCenter Integration Service extracts rows from effective dated PeopleSoft records according to the transformation options you define:

Number of Effective Dated Records in the Application Source Qualifier	Extract Current Rows Specified	Effective Date Join Order Specified	PowerCenter Integration Service Results
One	No	-	Ignores Extract Date. Returns all rows.
One	Yes	-	Uses Extract Date. Returns current rows for each key.
Multiple	Yes	Yes	Uses Extract Date. For each key, joins effective dated records based on the join order specified.
Multiple	Yes	No	Uses Extract Date. For each key, joins current rows from all records.

Number of Effective Dated Records in the Application Source Qualifier	Extract Current Rows Specified	Effective Date Join Order Specified	PowerCenter Integration Service Results
Multiple	No	Yes	Ignores Extract Date. For each key, joins effective dated records based on the join order specified.
Multiple	No	No	Ignores Extract Date. For each key, joins all rows from all records.

Handling EFFSEQ when Extracting Current Rows

Some effective dated PeopleSoft records include effective sequence columns. The effective sequence column, EFFSEQ, is used when multiple rows become effective on the same date. When you extract current rows from the effective dated records, the PowerCenter Integration Service selects the row with the most recent date in the effective sequence.

Examples of Joining Effective Dated Records

The following examples show how the PowerCenter Integration Service joins effective dated records:

- Extracting current rows with a join order
- Extracting current rows without a join order
- Joining effective dated records with a join order
- Joining effective dated records without a join order

The examples do not include effective sequences.

Each example includes the following information:

- The selected transformation options.
- What the PowerCenter Integration Service does.
- Results when you join two effective dated records.

The following table describes the examples you can use:

Term	Definition
Current Date	Date from which an effective date must be less than or equal to. When you specify a join order, the Extract Date is always the current date for the first table in the join order.
Effective Date	Value in the EFFDT column for a row of data.
Current Effective Date	Effective date that is the maximum date less than or equal to the current date.

The following examples use the sample source tables:

- Salary_data
- Title_data

The following table shows sample data with the EFFDT column in the Salary_data table:

EFFDT	EMP_ID	SALARY
1999-01-01 00:00:00.000	1000	45,000.00
2000-01-01 00:00:00.000	1000	52,000.00
2001-01-01 00:00:00.000	1000	65,000.00

The following table shows sample data with the EFFDT column in the Title_data table:

EFFDT	EMP_ID	TITLE
1999-01-01 00:00:00.000	1000	Associate
2000-06-30 00:00:00.000	1000	Sr. Associate

Extracting Current Rows with a Join Order

The following example shows how the PowerCenter Integration Service joins two effective dated PeopleSoft records when you define the following properties:

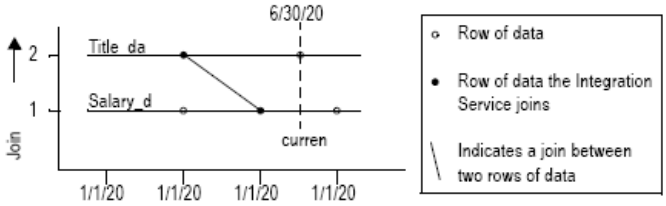
- Extract Current Rows = Yes
- Join Order = Salary_data, Title_data

When you extract current rows and designate a join order, the PowerCenter Integration Service performs the following tasks:

1. It uses the date in Extract Date as the current date for the first table.
2. It selects the row with the current effective date from the first table.
3. It uses the current effective date from the row of the first table *as the current date* for the second table.
4. It joins the row with the current effective date from the second table with the row from the first table.

Note: The PowerCenter Integration Service works similarly when you join more than two effective dated PeopleSoft records. The current effective date for the row of the second table becomes the current date for the third table in the join order.

The following example shows how the PowerCenter Integration Service joins the Salary_data and Title_data tables when you use Join Order = Salary_data, Title_data, and where the Extract Date is June 30, 2004:



The PowerCenter Integration Service uses June 30, 2004 as the current date for the first table, Salary_data. Then it selects the row from Salary_data with the current effective date. The PowerCenter Integration Service selects a row from the Title_data table using the following condition:

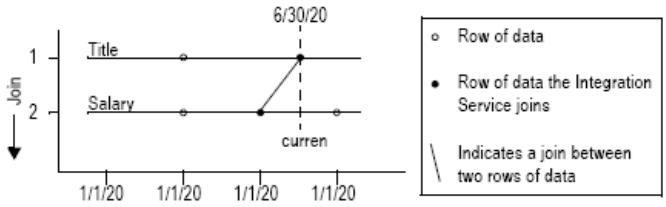
```
MAX(Title_data.EFFDT <= Salary_data.EFFDT)
```

The following table shows the data that the PowerCenter Integration Service returns when it joins the Salary_data and Title_data tables:

EMP_ID	SALARY	TITLE
1000	52,000.00	Associate

However, if you reverse the join order (Join Order = Title_data, Salary_data), the PowerCenter Integration Service returns different data.

The following figure shows how the PowerCenter Integration Service joins the Salary_data and Title_data tables when you use Join Order = Title_data, Salary_data, and where the Extract Date is June 30, 2004:



The PowerCenter Integration Service uses June 30, 2004 as the current date for the first table, Title_data. Then it selects the row from Title_data with the current effective date. Because a join order is specified, the PowerCenter Integration Service selects a row from the Salary_data table using the following condition:

```
MAX(Salary_data.EFFDT <= Title_data.EFFDT)
```

The following table shows the data that the PowerCenter Integration Service returns when it joins the Salary_data and Title_data tables:

EMP_ID	SALARY	TITLE
1000	52,000.00	Sr. Associate

Extracting Current Rows Without a Join Order

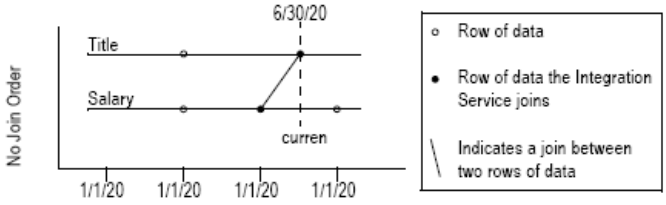
The following example shows how the PowerCenter Integration Service joins two effective dated PeopleSoft records when you define the following properties:

- Extract Current Rows = Yes
- Join Order = No

When you extract current rows and do not designate a join order, the PowerCenter Integration Service performs the following tasks:

1. It uses the date in Extract Date as the current date for all tables.
2. It selects the row with the current effective date from each table.
3. It joins the rows with the current effective date from all tables.

The following figure shows how the PowerCenter Integration Service joins the Salary_data and Title_data tables. The Extract Date is June 30, 2004:



The PowerCenter Integration Service uses June 30, 2004 as the current date for *both* tables. It selects the row from the Salary_data table with the current effective date. It selects the row from the Title_data table with the current effective date. It joins the rows from the Salary_data and Title_data tables.

The following table shows the row that the PowerCenter Integration Service returns when it joins the Salary_data and Title_data tables:

EMP_ID	SALARY	TITLE
1000	52,000.00	Sr. Associate

Joining Effective Dated Records with a Join Order

The following example shows how the PowerCenter Integration Service joins two effective dated PeopleSoft records when you define the following properties:

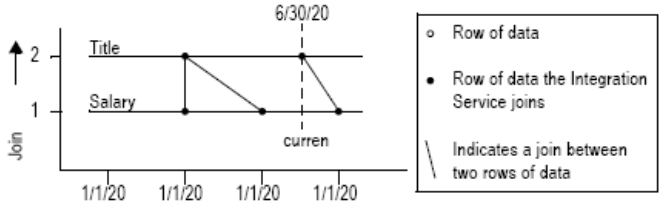
- Extract Current Rows = No
- Join Order = Salary_data, Title_data

When you do not extract current rows and define the join order, the PowerCenter Integration Service performs the following tasks:

1. It ignores the value in Extract Date.
2. It selects a row from the first table.
3. It uses the current effective date from the row of the first table as the current date for the second table.
4. It joins the row with the current effective date from the second table with the row from the first table.
5. It repeats the steps for all rows in the first table.

Note: The PowerCenter Integration Service works similarly when you join more than two effective dated PeopleSoft records. The current effective date for the row in the second table becomes the current date for the third table in the join order.

The following figure shows how the PowerCenter Integration Service joins the Salary_data and Title_data tables, where the current date is June 30, 2004:



The PowerCenter Integration Service ignores the current date. It selects each row from the Salary_data table and joins each one with a row from the Title_data table. It joins the row from Salary_data with a row from Title_data using the following condition:

```
Title_data.EFFDT <= Salary_data.EFFDT
```

The following table shows the rows that the PowerCenter Integration Service returns when it the Salary_data and Title_data tables:

EMP_ID	SALARY	TITLE
1000	45,000.00	Associate
1000	52,000.00	Associate
1000	65,000.00	Sr. Associate

Joining Effective Dated Records Without a Join Order

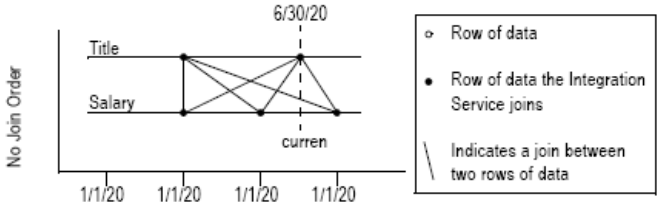
The following example shows how the PowerCenter Integration Service joins two effective dated PeopleSoft records when you define the following properties:

- Extract Current Rows = No
- Join Order = No

When you do not extract current rows or define a join order, the PowerCenter Integration Service performs the following tasks:

1. It ignores the value in Extract Date.
2. It joins each row in the first table with all rows from all other tables.

The following figure shows how the PowerCenter Integration Service joins the Salary_data and Title_data tables, and where the current date is June 30, 2004:



The PowerCenter Integration Service ignores the current date. It performs a full outer join. It selects each row from the Sample_data table and joins each one with all rows from the Title_data table.

The following table shows the rows that the PowerCenter Integration Service returns when it joins Salary_data and Title_data tables:

EMP_ID	SALARY	TITLE
1000	45,000.00	Associate
1000	45,000.00	Sr. Associate

EMP_ID	SALARY	TITLE
1000	52,000.00	Associate
1000	52,000.00	Sr. Associate
1000	65,000.00	Associate
1000	65,000.00	Sr. Associate

Configuring an Application Source Qualifier

By default, the Designer creates an Application Source Qualifier when you drag a PeopleSoft source definition into the Mapping Designer. Do not edit the datatypes in the Application Source Qualifier. The datatypes determine how the source database binds data when you import it. The datatypes in the source definition must correspond with the datatypes in the Application Source Qualifier. Otherwise, the mapping is invalid.

To configure an Application Source Qualifier:

1. In the Mapping Developer, double-click the title bar of an Application Source Qualifier.
2. Click the Properties tab.
3. Optionally, enter the following settings:

Option	PeopleSoft Source Definition Types	Description
Filter	Records	Filter conditions the PowerCenter Integration Service applies when extracting source rows. When you enter a filter, the Designer adds the filter to the default query. Use syntax supported by the source database. Enter only the filter condition, do not type WHERE.
Join Override	Records	Join conditions to join data from connected sources. Overrides the default join. When you enter a Join Override, the Designer adds the join to a WHERE clause in the default query. Use syntax supported by the source database. Enter only join conditions. Do not enter WHERE.
Extract Override	Records	Overrides the default query. Specifies the query you want the PowerCenter Integration Service to use to extract data from PeopleSoft records. Enter the entire extract query with syntax supported by the source database. If you enter an Extract Override, the Designer ignores any changes you make to transformation properties. The PowerCenter Integration Service uses the extract override to query sources. It does not perform language code queries. Use for PeopleSoft records only. If you connect the Application Source Qualifier to a PeopleSoft tree, the PowerCenter Integration Service ignores the extract override. It uses the default query.

Option	PeopleSoft Source Definition Types	Description
Number of Sorted Ports	Records	Number of columns from a PeopleSoft record the PowerCenter Integration Service sorts. If you select this option, the Designer adds an ORDER BY to the default query and sorts the specified number of columns, beginning at the top of the Application Source Qualifier. Use for PeopleSoft records only. The Designer ignores PeopleSoft tree ports connected to the Application Source Qualifier.
Tracing Level	Records, Imported Trees, Created Tree Source Definitions	Amount of detail included in the session log when you use a session containing this transformation.
Select Distinct	Records	Selects only unique source rows. If you choose this option, the Designer uses SELECT DISTINCT in the default query.
Extract Current Rows	Records	Extracts the current rows from effective dated records. Uses the date in Extract Date as the current date. When you use this option, the Designer modifies the default query and default join to select only the most recent data as defined by the extract date. When you use this option, the PowerCenter Integration Service extracts only one row for each key. Use for effective dated PeopleSoft records only.
Extract Date	Records	Current date to use with the Extract Current Rows option. To use the current date, leave this attribute empty. Enter the date in the following format: `MM/DD/YYYY HH24:MI:SS` Use for effective dated PeopleSoft records only.
Effective Date Join Order	Records	Join order between effective dated records. When you use this option, the PowerCenter Integration Service joins a row from the second table with a row from the first table. The effective date of the row from the second table is the maximum date less than or equal to the effective date of the row from the first table. Use for multiple effective dated PeopleSoft records only.

4. Click Sources and indicate additional source definitions to associate with this Application Source Qualifier.

   Only identify associated sources when you plan to join data from multiple tables or similar application systems.

5. From the Metadata Extensions tab, create user-defined metadata extensions.

   Use metadata extensions to associate information with repository metadata.

6. Click OK.

CHAPTER 5

Accessing XLATTABLE Data

This chapter includes the following topics:

- Accessing XLATTABLE Data Overview, 60
- Sourcing XLATTABLE Data, 61
- Looking Up XLATTABLE Data, 63

Accessing XLATTABLE Data Overview

PeopleSoft systems use the XLATTABLE table to define codes used in PeopleSoft records. Use PowerExchange for PeopleSoft to access XLATTABLE data to define those codes. For example, a PeopleSoft record might have a row with “A” in the STATUS column. By accessing the XLATTABLE table, you can determine that A stands for “Available.”

In a mapping, you can access the XLATTABLE table in two ways:

- **As a PeopleSoft source.** You can import the XLATTABLE table as a PeopleSoft source definition and then join it with a PeopleSoft record in an Application Source Qualifier.

  When you join source definitions in the Application Source Qualifier, the PowerCenter Integration Service performs an inner equijoin. Use the XLATTABLE table as a source when all source rows in the PeopleSoft record have corresponding records in the XLATTABLE table.
- **As a database lookup table.** You can also import the XLATTABLE table from the underlying database of the PeopleSoft system. As a relational table, use the XLATTABLE table in a Lookup transformation. You can create Lookup transformations based on PeopleSoft sources.

  As a Lookup transformation, you can look up corresponding data in the XLATTABLE table. Use this option to look up more than one column or to perform more complicated queries against the XLATTABLE table, such as choosing the first or last results of a multiple match or replacing null values.

Locating Accurate Values

To ensure the PowerCenter Integration Service extracts accurate information from the XLATTABLE table, each time you use the XLATTABLE table, enter the following information:

- A condition to indicate which columns to compare
- A filter to remove unrelated code definitions and definitions in unnecessary languages

When you use the XLATTABLE table as a PeopleSoft source, enter a join override and filter conditions to extract XLATTABLE data.

When you use the XLATTABLE table as a lookup table, enter lookup conditions and a lookup SQL override to extract XLATTABLE data.

Codes defined in the XLATTABLE table can contain more than one definition for a given code. For example, “A” might stand for “Active” in one PeopleSoft record, and it might stand for “Assets” in another. To differentiate between different codes, the XLATTABLE table contains two columns that are directly referenced by PeopleSoft records:

- **FIELDVALUE.** Contains codes defined within the XLATTABLE table and used by PeopleSoft records.
- **FIELDNAME.** Contains a text string. The text string is the name of the column in the related PeopleSoft record that contains the code. For example, the code “A” is in the STATUS column of a PeopleSoft record. The corresponding row in the XLATTABLE table has “A” in the FIELDVALUE column and “STATUS” in the FIELDNAME column.

Additional columns define the codes, including the following columns:

- **LANGUAGE_CD.** PeopleSoft language code. You can define a code in the XLATTABLE table in different languages. The language code for each row indicates the language of the description. When the XLATTABLE table defines codes in different languages, you can add a filter condition to extract data in the language you want.
- **EFFDT.** Effective date of the row.
- **XLATLONGNAME.** A long description of the code.
- **XLATSHORTNAME.** A short description of the code.

For example, a PeopleSoft record contains a column named BD_WKBOOK_TYPE. The column contains codes representing different workbook types, such as A and E.

The following table shows the codes defined in the XLATTABLE table:

FIELDNAME	FIELDVALUE	EFFDT	XLATLONGNAME	XLATSHORTNAME
BD_WKBOOK_TYPE	A	01-JAN-00	Assets Workbook	Assets
BD_WKBOOK_TYPE	E	01-JAN-00	Expenses Workbook	Expenses
BD_WKBOOK_TYPE	P	01-JAN-00	Payroll Workbook	Payroll
BD_WKBOOK_TYPE	R	01-JAN-00	Revenue Workbook	Revenue

In the XLATTABLE table, “BD_WKBOOK_TYPE” identifies the codes in the FIELDVALUE column. Definitions of these codes appear in the XLATSHORTNAME and XLATLONGNAME columns. The short description of the “A” workbook type is “Assets.” The long description is “Assets Workbook.”

Sourcing XLATTABLE Data

You can access XLATTABLE data as a PeopleSoft source and join it to a related PeopleSoft record in a single Application Source Qualifier transformation. The join type in an Application Source Qualifier is an inner equijoin. With an inner equijoin, the PowerCenter Integration Service extracts data from one source table when it finds a matching row in the other source table. It does not extract rows without matching rows. If you want the PowerCenter Integration Service to extract all rows from a PeopleSoft record, regardless of whether the matching row appears in the XLATTABLE table, use the XLATTABLE table as a lookup table.

Complete the following steps to use the XLATTABLE table as a source:

1. Import the XLATTABLE table as a PeopleSoft source definition.
2. Add the XLATTABLE source definition to the mapping.
3. Override the default query in the Application Source Qualifier transformation.
4. Configure the database connection and run the workflow.

Step 1. Import the XLATTABLE Table from PeopleSoft

To use XLATTABLE data as a source, import it as a PeopleSoft source definition. In the Import from PeopleSoft dialog box, the XLATTABLE table appears in the list of available PeopleSoft records.

Tip: Use the import filter to reduce the number of PeopleSoft records the Designer displays.

Step 2. Add the XLATTABLE Source Definition to the Mapping

After importing the XLATTABLE table as a PeopleSoft source definition, you can add it to a mapping. To join XLATTABLE data with a PeopleSoft record, connect both source definitions to the same Application Source Qualifier transformation.

Step 3. Override the Default Query

To extract the data you need from the XLATTABLE table, enter a join override and filter conditions in the Application Source Qualifier transformation. The join override allows the PowerCenter Integration Service to compare the two source columns with codes: the column containing codes in the PeopleSoft record with the XLATTABLE FIELDVALUE column. The filter ensures the PowerCenter Integration Service extracts only data from the XLATTABLE table that relates to the PeopleSoft record.

Entering the XLATTABLE Join Condition

In the join override, set the XLATTABLE FIELDVALUE column equal to the column in the PeopleSoft record that contains the codes you want to describe:

```
XLATTABLE.FIELDVALUE=[RECORD_NAME].[RECORD_COLUMN_NAME]
```

For example, enter:

```
XLATTABLE.FIELDVALUE=PS_ALLOC_BASWK_TBL.DEPTID
```

Entering the XLATTABLE Source Filter

You can enter several filter conditions in an Application Source Qualifier transformation. To extract accurate XLATTABLE values, enter the following filter conditions:

```
XLATTABLE.FIELDNAME = '[RECORD_COLUMN_NAME]' AND LANGUAGE_CD = '[LANGUAGE_CODE]'
```

The first condition sets the XLATTABLE FIELDNAME column equal to the column name in the PeopleSoft record that contains the codes you want to define. Use this clause to ensure the PowerCenter Integration Service returns descriptions for the codes used in the mapping.

In the second condition, enter a PeopleSoft language code. This determines the language in which the PowerCenter Integration Service extracts language-sensitive data. This language code must be identical to the language code used in the application connection. If you do not enter a language code in the application connection, the language code in the filter condition must correspond to the language of the PeopleSoft record base table.

If necessary, you can add additional conditions, such as:

```
EFF_STATUS = '[ROW_EFFECTIVE_STATUS]' AND EFFDT = [ROW_EFFECTIVE_DATE]
```

Use any SQL statements supported by the underlying database of the PeopleSoft system.

For example, the ALLOC_BASWK_TBL table contains department IDs that are described in the XLATTABLE table. You want to define the codes in the DEPT_ID column, and you want language-sensitive data in English. Enter the necessary filter conditions in the Application Source Qualifier transformation. For example, enter the following filter conditions for the XLATTABLE table:

```
XLATTABLE.FIELDNAME= 'DEPT ID' AND XLATTABLE.LANGUAGE_CD= 'ENG'
```

To ensure language-sensitive data matches, when you configure the application connection for this mapping, set the language code to ENG.

Step 4. Configure the Application Connection and Run Workflows

When you create an application connection for the session, enter the language code used in the filter condition for the Application Source Qualifier. For example, if you enter the ENG language code in the Application Source Qualifier, enter the ENG language code in the application connection.

When you create a session for a mapping containing an XLATTABLE source table, you configure it as you would any other PeopleSoft session. If you import the XLATTABLE table with a database username that is not the owner of the table, enter a table name prefix in the session.

Looking Up XLATTABLE Data

You can access the XLATTABLE table as lookup table by importing it from the underlying database of the PeopleSoft system. Use a Lookup transformation to look up XLATTABLE data from the relational table. You cannot use a Lookup transformation to look up data in a PeopleSoft source.

To extract accurate XLATTABLE data, configure a lookup SQL override in the Lookup transformation. This requires caching the lookup table.

The PowerCenter Integration Service looks up data in the XLATTABLE table and returns data as defined in a lookup condition. By default, the PowerCenter Integration Service returns NULL when it finds no matching conditions. However, you can configure the transformation to return a default value instead.

Complete the following steps to look up XLATTABLE data:

1. Import the XLATTABLE table as a database table.
2. Create and configure a Lookup transformation.
3. Configure a PeopleSoft application connection.
4. Enter XLATTABLE table location information in the mapping or session.

Use either a connected or unconnected Lookup transformation to access XLATTABLE data.

Step 1. Import the XLATTABLE Table from the Database

To use the PeopleSoft XLATTABLE table as a Lookup transformation in a mapping, import it as a *relational* table. You can only create a Lookup transformation for relational tables.

You can import the XLATTABLE table from the database when you create the Lookup transformation. You can also import the XLATTABLE table as a source definition and reference the source definition in the repository for the Lookup transformation.

When you import the XLATTABLE table as a source definition, you import it from the PeopleSoft system underlying database. In the Import from Database dialog box, use the table owner name and password or a database username with SELECT permission on the table. To keep XLATTABLE data secure, do not use the table owner name to import the source definition.

To import the XLATTABLE table:

1. In the Source Analyzer, click Sources > Import from Database.
2. Select the ODBC data source you use to connect to the PeopleSoft underlying database, enter a database username and password.

   The database username and password must have SELECT permission on the XLATTABLE table.
3. If the database username entered is not the owner of the table, enter the table owner name. Click Connect.
4. In the Tables window, open the Table folder, select XLATTABLE, and click OK.

   You can now create a Lookup transformation based on the XLATTABLE table.

Step 2. Create and Configure the Lookup Transformation

To look up XLATTABLE data, create a Lookup transformation in the mapping.

In the Lookup transformation, configure the following properties:

- **Lookup condition.** Allows the PowerCenter Integration Service to compare the input column containing codes with the XLATTABLE FIELDVALUE column.
- **Lookup SQL override.** Ensures the PowerCenter Integration Service extracts only XLATTABLE data that relates to input data.
- **Lookup cache.** Allows the PowerCenter Integration Service to perform a lookup SQL override.

Defining the Lookup Condition

To access XLATTABLE data through a Lookup transformation, define the lookup condition. Before defining the lookup condition, create an input port for the data you want to pass to the Lookup transformation. This input port contains the codes you want to describe. Use this port in the lookup condition. The PowerCenter Integration Service compares this data with corresponding data in the XLATTABLE table.

The lookup condition sets the FIELDVALUE port of the XLATTABLE table equal to the input port you create:

```
FIELDVALUE = INPUT_PORT
```

For example, you create a mapping that looks up the XLATSHORTNAME value from the XLATTABLE table to define customer status codes. The Lookup transformation contains an input port, CUST_STATUS_IN, to accept customer status codes from the Source Qualifier. These codes include A and I.

The following table shows the rows that can be defined in the XLATTABLE table:

FIELDNAME	FIELDVALUE	EFFDT	XLATLONGNAME	XLATSHORTN
CUST_STATUS	A	01-JAN-2000	Active	Active
CUST_STATUS	I	01-JAN-2000	Inactive	Inactive

The following lookup condition in LKP_XLATTABLE Lookup transformation looks up the customer status definitions in the XLATTABLE table:

```
FIELDVALUE = CUST_STATUS_IN
```

The Lookup Table Column value is FIELDVALUE. The Transformation Port value CUST_STATUS_IN contains customer status codes.

Entering the Lookup SQL Override

When you create a Lookup transformation to access XLATTABLE data, enter a lookup SQL override to filter out unrelated XLATTABLE data. Before you create a lookup SQL override, connect transformation output ports to the rest of the mapping. The Designer uses these connections to generate the default lookup SQL statement.

To enter a lookup SQL override, generate the default SQL. Then, add a WHERE clause to the end of the default statement. The clause must contain the following:

```
WHERE FIELDNAME = '[INPUT_PORT_NAME]' AND LANGUAGE_CD = '[LANGUAGE_CODE]'
```

In the first condition, INPUT_PORT_NAME is the name of the input port used in the lookup condition. This port contains the codes you want to describe. Use this clause to ensure the PowerCenter Integration Service returns descriptions for the codes used in the mapping.

In the second condition, enter the PeopleSoft language code. This determines the language in which the PowerCenter Integration Service extracts language-sensitive data. This language code must be identical to the language code used in the session database connection. If you do not enter a language code in the session database connection, the language code in the lookup SQL override must correspond to the language of the PeopleSoft record base table.

If necessary, you can add additional conditions, such as:

```
EFF_STATUS = '[ROW_EFFECTIVE_STATUS]' AND EFFDT = [ROW_EFFECTIVE_DATE]
```

Use any SQL statements supported by the underlying database of the PeopleSoft system.

The Designer generates the following SQL statement for the Lookup transformation LKP_XLATTABLE:

```
SELECT XLATTABLE.FIELDNAME as FIELDNAME, XLATTABLE.LANGUAGE_CD as LANGUAGE_CD,
XLATTABLE.EFFDT as EFFDT, XLATTABLE.VERSION as VERSION, XLATTABLE.EFF_STATUS as
EFF_STATUS, XLATTABLE.XLATLONGNAME as XLATLONGNAME, XLATTABLE.XLATSHORTNAME as
XLATSHORTNAME, XLATTABLE.FIELDVALUE as FIELDVALUE FROM XLATTABLE
```

You can then add a WHERE clause with additional conditions as follows:

```
WHERE FIELDNAME = 'CUST_STATUS' AND LANGUAGE_CD = 'ENG' AND EFF_STATUS = 'A'

AND EFFDT = (SELECT MAX(EFFDT) FROM XLATTABLE SECONDARY_NAME WHERE

FIELDNAME = 'CUST_STATUS' AND EFF_STATUS = 'A' AND LANGUAGE_CD = 'ENG' AND

SECONDARY_NAME.FIELDVALUE = XLATTABLE.FIELDVALUE)
```

This clause ensures the PowerCenter Integration Service returns values from the XLATTABLE table where the FIELDNAME is “CUST_STATUS,” the language is English, the status is active, and the effective date is the most recent. This statement uses “SECONDARY_NAME” as a secondary reference for the XLATTABLE table to allow a self-join.

Note: Do not edit the existing syntax of the default lookup SQL statement.

Adding the Owner Name

When you import the XLATTABLE table using a database username that is *not* the owner of the table, enter the table owner name in the lookup SQL override. When you enter the table owner name to the SQL

statement, you define the XLATTABLE table as "*ownername*.XLATTABLE." To ensure the lookup SQL override remains valid, you must also rename the table XLATTABLE.

For example, the owner of the XLATTABLE table used in the Lookup transformation LKP_XLATTABLE is *jdoe*. You import the table using a different database username.

When you enter the lookup SQL override, first generate the default SQL:

```
SELECT XLATTABLE.FIELDNAME as FIELDNAME, XLATTABLE.LANGUAGE_CD as LANGUAGE_CD,
XLATTABLE_EFFDT as EFFDT, XLATTABLE.VERSION as VERSION, XLATTABLE.EFF_STATUS as
EFF_STATUS, XLATTABLE.XLATLONGNAME as XLATLONGNAME, XLATTABLE.XLATSHORTNAME as
XLATSHORTNAME, XLATTABLE.FIELDVALUE as FIELDVALUE FROM XLATTABLE
```

To enter the XLATTABLE table owner name and rename the table XLATTABLE, edit the table name as follows:

```
FROM jdoe.XLATTABLE XLATTABLE
```

You can then add a WHERE clause and additional conditions as follows:

```
WHERE FIELDNAME = 'CUST_STATUS' AND LANGUAGE_CD = 'ENG' AND EFF_STATUS = 'A' AND

EFFDT = (SELECT MAX(EFFDT) FROM jdoe.XLATTABLE SECONDARY_NAME WHERE

FIELDNAME = 'CUST_STATUS' AND EFF_STATUS = 'A' AND LANGUAGE_CD = 'ENG' AND

SECONDARY_NAME.FIELDVALUE = XLATTABLE.FIELDVALUE)
```

As in the previous example, this clause ensures the PowerCenter Integration Service returns values from the XLATTABLE table where the FIELDNAME is "CUST_STATUS," the language is English, the status is active, and the effective date is the most recent. This statement uses "SECONDARY_NAME" as a secondary reference for the XLATTABLE table to allow a self-join. It also includes the table owner name "jdoe.XLATTABLE."

The complete lookup SQL override for the Lookup transformation is as follows:

```
SELECT XLATTABLE.FIELDNAME as FIELDNAME, XLATTABLE.LANGUAGE_CD as LANGUAGE_CD,
XLATTABLE_EFFDT as EFFDT, XLATTABLE.VERSION as VERSION, XLATTABLE.EFF_STATUS as
EFF_STATUS, XLATTABLE.XLATLONGNAME as XLATLONGNAME, XLATTABLE.XLATSHORTNAME as
XLATSHORTNAME, XLATTABLE.FIELDVALUE as FIELDVALUE FROM jdoe.XLATTABLE XLATTABLE WHERE
FIELDNAME = 'CUST_STATUS' AND LANGUAGE_CD = 'ENG' AND EFF_STATUS = 'A' AND

EFFDT = (SELECT MAX(EFFDT) FROM jdoe.XLATTABLE SECONDARY_NAME WHERE

FIELDNAME = 'CUST_STATUS' AND EFF_STATUS = 'A' AND LANGUAGE_CD = 'ENG' AND
SECONDARY_NAME.FIELDVALUE = XLATTABLE.FIELDVALUE)
```

Validating Lookup SQL Override Syntax

After you enter the lookup SQL override, you can validate the syntax of the statement.

To validate lookup SQL override syntax:

1. In the SQL Editor, select the ODBC data source to connect to the XLATTABLE table.

   This data source connects to the PeopleSoft underlying database.
2. Enter a database username and password to connect to the database and click Validate.

Caching the Lookup Table

When you use a lookup SQL override, you must also cache the lookup table. Lookup transformations cache the lookup table by default.

If the data in the XLATTABLE is static, you can configure the Lookup transformation for a persistent lookup cache. This allows the PowerCenter Integration Service to save the lookup cache to file and reuse data in the file the next time you run the session. If you use a persistent lookup cache, you can optionally define a name for the lookup cache files.

Step 3. Configure a PeopleSoft Application Connection

When you create an application connection for the workflow, enter the language code used in the lookup SQL override. For example, enter the ENG language code in the application connection.

Step 4. Enter Location Information in the Mapping or Session

When using a Lookup transformation to access XLATTABLE data, configure the Location Information property on the Properties tab of the Lookup transformation. The PowerCenter Integration Service uses information in the Location Information property to locate the XLATTABLE table.

By default, the Location Information value is $Source for the XLATTABLE table, since you import it from a source database. When you accept the default, the PowerCenter Integration Service uses the session source application connection to locate the XLATTABLE table. You can select a database connection to replace the default location.

The database connection the PowerCenter Integration Service uses to access the XLATTABLE table must have the same database username, password, and connect string as the ODBC data source used to import the table.

If the session database connection does not fit this criteria, select a database connection for the Lookup transformation in the Lookup transformation or in the session properties.

CHAPTER 6

Creating PeopleSoft Sessions and Workflows

This chapter includes the following topics:

- Configuring a Session for PeopleSoft Sources, 68
- Scheduling a Workflow, 69

Configuring a Session for PeopleSoft Sources

You configure different application connections depending if the session extracts data from PeopleSoft-only sources or from heterogeneous sources.

Configuring a Session with PeopleSoft-only Sources

When you create a session using PeopleSoft source data, ensure the session is configured to connect to the PeopleSoft source system.

Note: The PowerCenter Integration Service does not perform extract overrides when an Application Source Qualifier is connected to a PeopleSoft tree. When the PowerCenter Integration Service ignores an extract override, it notes the change in the session log.

To configure a session with PeopleSoft-only sources:

1. In the Task Developer, double-click a PeopleSoft session to open the session properties.

   The Edit Tasks dialog box appears.
2. Click the Mapping tab.

   If the mapping contains a PeopleSoft source definition, the session uses Application for the source type by default
3. Choose a value for the application connection.

   Use the application connection for the underlying database containing PeopleSoft data. For example, if the PeopleSoft system is on an Oracle database, select the application connection configured for the Oracle database.

Configuring a Session with Heterogeneous Sources

If the session contains both PeopleSoft and non-PeopleSoft sources, you need to configure a heterogeneous PeopleSoft session and connections for each source.

To configure a session with heterogeneous sources:

1. Select the Mapping tab in the session properties.
2. Select the appropriate connection type for each source.

   As with a non-heterogeneous PeopleSoft session, use the application connection for the PeopleSoft underlying database to access the PeopleSoft source.

Entering a Source Table Owner Name

The database user in the application connection must have SELECT permission on the source tables accessed in the session. If this database user is not the owner of the source tables, enter the table owner name in the session as a source table prefix.

Note: If the mapping contains a Source Qualifier with an SQL Override or an Application Source Qualifier with an extract override, the PowerCenter Integration Service ignores the table name prefix setting for all connected sources.

To enter a source table prefix in the session:

1. In the session properties, click a target on the Mapping tab.
2. Enter the owner name in the Table Name Prefix field for each listed table.
3. Click OK.

Configuring a PeopleSoft Session to Partition Data

If you need to extract a large amount of source data, you can partition the sources to improve session performance. Partitioning sources allows the PowerCenter Integration Service to create multiple connections to sources and process partitions of source data concurrently. You can partition sources if the PowerCenter Integration Service can maintain data consistency when it processes the partitioned data.

By default, the Workflow Manager sets the partition type to pass-through for PeopleSoft trees. In pass-through partitioning, the PowerCenter Integration Service passes all rows at one partition point to the next partition point without redistributing them. You can also specify key range partitioning for PeopleSoft trees. In key range partitioning, the PowerCenter Integration Service passes data through each partition depending on the ranges specified for each port.

You cannot create multiple partitions on an Application Source Qualifier for PeopleSoft when it is connected to or associated with a PeopleSoft tree.

Scheduling a Workflow

Before you run a workflow, configure and schedule the workflow. You can schedule a non-reusable scheduler for a workflow. Or, you can create a reusable scheduler to use with the workflow.

You can schedule a workflow to run continuously, run at a given time or interval, or you can manually start a workflow. The PowerCenter Integration Service runs scheduled workflows through the duration of the schedule unless the workflow fails.

APPENDIX A

Datatype Reference

This appendix includes the following topic:

- PeopleSoft and Transformation Datatypes, 70

PeopleSoft and Transformation Datatypes

PowerCenter uses the following datatypes in PeopleSoft mappings:

- **PeopleSoft native datatypes.** PeopleSoft datatypes appear in PeopleSoft definitions in a mapping.
- **Transformation datatypes.** Set of datatypes that appear in the transformations. They are internal datatypes based on ANSI SQL-92 generic datatypes, which the PowerCenter Integration Service uses to move data across platforms. They appear in all transformations in a mapping.

When the PowerCenter Integration Service reads source data, it converts the native datatypes to the comparable transformation datatypes before transforming the data. When the PowerCenter Integration Service writes to a target, it converts the transformation datatypes to the comparable native datatypes.

The following table lists the PeopleSoft datatypes that PowerCenter supports and the corresponding transformation datatypes:

PeopleSoft Datatype	Range	Transformation Datatype	Range
Character	1 to 254 characters.	String or Nstring	1 to 104,857,600 characters. Fixed-length or varying-length string.
ContentReference	1 to 30 characters.	String or Nstring	1 to 104,857,600 characters. Fixed-length or varying-length string.
Date	Field length of 10, four-digit year.	Date/Time	Jan 1, 0001 A.D. to Dec 31, 9999 A.D. Precision to the nanosecond.
DateTime	Field length of 26, maximum precision is database-dependent; format is user-specified.	Date/Time	Jan 1, 0001 A.D. to Dec 31, 9999 A.D. Precision to the nanosecond.
Image	Database dependent.	Not supported	-
Long Character	1 to 65,535 characters, based on database environment.	Text or Ntext	1 to 104,857,600 characters. Fixed-length or varying-length string.

PeopleSoft Datatype	Range	Transformation Datatype	Range
Long Character Raw	Database dependent.	Binary	1 to 104,857,600 bytes. You can pass binary data from a source to a target, but you cannot perform transformations on binary data.
Number	Precision 1 to 18.	Decimal	Precision 1 to 28.
Signed Number	Precision 1 to 18.	Decimal	Precision 1 to 28.
SubRecord	Each field in a SubRecord has the range of the declared datatype.	-	-
Time	Date.	Date/Time	Jan 1, 0001 A.D. to Dec 31, 9999 A.D. Precision to the nanosecond.
Version	Database dependent.	Not supported	-

Unsupported Datatypes

PowerExchange for PeopleSoft does not support the following PeopleSoft datatypes:

- **Image.** When you import a PeopleSoft source definition with an Image column, the Designer imports the column. The PowerCenter Integration Service replaces Image fields with NULL.
- **Version.** When you import a PeopleSoft source definition with a Version column, the Designer imports the source definition without the Version column. The PowerCenter Integration Service does not extract data from the Version columns.

PeopleSoft Datatypes

PeopleSoft saves data in underlying database tables. Therefore, the type and range for PeopleSoft datatypes are dependent on the underlying database type. For example, when you have a PeopleSoft system on Microsoft SQL Server, the PeopleSoft Number datatype is saved to the Microsoft SQL Server table as a Decimal native datatype with a range of -10e38 to 10e38-1.

When you import a PeopleSoft source definition, the Designer connects to the PeopleSoft metadata tables to import PeopleSoft metadata. PeopleSoft metadata describes columns in terms of PeopleSoft datatypes, precision, and scale. As a result, PeopleSoft source definitions appear with PeopleSoft datatypes.

PeopleSoft SubRecords

When you import a PeopleSoft source that uses a SubRecord datatype, the Designer expands the SubRecord, importing each column represented by the SubRecord as an individual source column. You treat these columns as you would any other column.

A PeopleSoft SubRecord can contain multiple columns, each with its own datatype. For example, to use the columns FIRST_NAME, LAST_NAME, SS_NUM in multiple records, you might create a PeopleSoft SubRecord named PERSONAL that contains metadata for these three columns. You can then use the PERSONAL SubRecord in PeopleSoft objects rather than recreating the same three columns in each object. Thus, when you import the PERSONAL SubRecord, the Designer creates three columns: FIRST_NAME, LAST_NAME, and SS_NUM, each with their respective datatypes.

APPENDIX B

PeopleSoft Language Codes

This appendix includes the following topic:

- PeopleSoft Language Codes, 72

PeopleSoft Language Codes

In PeopleSoft, you create metadata using a base language specified in the PeopleSoft application. You can also include non-base language translations for language-sensitive metadata. Language-sensitive metadata can include column names, business names, descriptions, and comments.

When you import a PeopleSoft source definition, the Designer imports metadata in the base language by default. If you want language-sensitive metadata in a different language, enter a PeopleSoft language code. If the source has language-sensitive metadata for the language code, the Designer imports it. If the source does not have language-sensitive metadata for the language code, the Designer imports metadata in the base language.

For example, you want to import a source definition that uses French as its base language. You want to import the PeopleSoft records with English metadata. In the Import from PeopleSoft dialog box, enter the PeopleSoft language code ENG to import English translations. The Designer imports available English metadata for the records you select. If English translations are not available, the Designer imports French metadata.

By default, PeopleSoft stores data for each source in a base table. Typically, language-sensitive data in the base table is in the base language. PeopleSoft stores translations of language-sensitive data in a related language table.

By default, the PowerCenter Integration Service extracts all data for a PeopleSoft source from the base table. If you want language-sensitive data in a particular language, enter the corresponding PeopleSoft language code in the application connection for the PeopleSoft system.

The PowerCenter Integration Service queries both the base table and the related language table. When language-sensitive data exists for the specified language code, the PowerCenter Integration Service extracts the data. When data does not exist for the specified language code, the PowerCenter Integration Service extracts all data from the base table.

The following table lists common PeopleSoft language codes:

Language (Long Name)	Language (Short Name)	PeopleSoft Language Code
Canadian French	Can French	CFR
Dutch	Dutch	DUT
English	English	ENG
Spanish	Spanish	ESP
French	French	FRA
German	German	GER
International English	Intl Eng	INE
Japanese	Japanese	JPN
Portuguese	Portuguese	POR

The language codes associated with a PeopleSoft system might differ based on the PeopleSoft version and installation. For details specific to your installation, see the PeopleSoft documentation.

APPENDIX C

Glossary

branch

A unit of a tree. A branch is a logical unit of nodes that extends from the root node to the detail range. Branches allow skipped or missing nodes.

branch node

A node of a tree containing hierarchical information.

created tree source definition

A source definition you create in the Source Analyzer by choosing Import From PeopleSoft from the Sources menu. The PowerCenter Integration Service vertically flattens all created tree source definitions.

detail range

A numeric range associated with a node in a detail tree. Detail ranges represent ranges of key values in the related detail record of the tree.

detail record

A PeopleSoft record associated with a PeopleSoft tree. Detail records contain data identified by a PeopleSoft tree.

detail tree

A type of PeopleSoft tree that contains detail ranges.

EFFDT

The name of a column used by some PeopleSoft source tables to version data.

EFFSEQ

The name of a column used by some PeopleSoft source tables in conjunction with the EFFDT column to version data. Also used independently for general sequences.

flattening

See *horizontal flattening* on page 74 and *vertical flattening* on page 75.

horizontal flattening

A method of extracting data from PeopleSoft trees that results in a row for each unique branch and detail combination. See also *vertical flattening* on page 75.

loose-level tree

A type of PeopleSoft tree. In a loose-level tree, one level can contain nodes with different kinds of information. Also, nodes representing the same kind of information can appear at multiple levels. Loose-level trees can contain branches with skipped or missing nodes.

node

A unit of a tree. A node contains categorical or contextual information.

panel

A type of PeopleSoft metadata. Each panel can reference several PeopleSoft records.

PeopleSoft keys

A key type reserved for PeopleSoft sources that contain both unique and non-unique key columns.

record

A type of PeopleSoft metadata with a table-like logical structure. Like a relational table, a PeopleSoft record can contain columns with defined datatypes, precision, scale, and keys.

root node

The highest node in the structure of the hierarchy. It is the origin of all other nodes.

SetID

A value that provides a unique identifier for each hierarchy.

SQL table

A type of PeopleSoft record, similar to a database table.

SQL view

A type of PeopleSoft record, similar to a database view.

strict-level tree

A type of PeopleSoft tree. In a strict level tree, each branch progresses through a specific hierarchy of tree levels, from the root node to the detail range. Strict-level trees support branches with skipped or missing nodes as long as existing nodes do not violate the logical order of tree levels.

summary tree

A type of PeopleSoft tree. Summary trees provide an alternative view of nodes in a detail tree. Summary trees contain no detail ranges.

tree

A type of PeopleSoft metadata. A PeopleSoft tree defines a hierarchy of relationships for values in a single column of data in the related PeopleSoft record known as a *detail record*.

vertical flattening

A method of extracting data from PeopleSoft trees that results in a row for each node or detail range represented in the tree. See also *horizontal flattening* on page 74.

vertical tree

See *created tree source definition* on page 74.

winter tree

A type of PeopleSoft tree. A winter tree extracts data from loose-level and strict-level node-oriented trees. It contains no detail ranges or detail records.

XLATTABLE table

A PeopleSoft system table used to store data referenced by PeopleSoft records.

INDEX

A

application connections
  PeopleSoft language codes 72
  PeopleSoft username 69
Application Source Qualifier
  default join for PeopleSoft records 40
  default query for PeopleSoft records 38, 39
  Extract Override property for PeopleSoft records 47
  extracting current rows from PeopleSoft records 54
  filtering data from PeopleSoft sources 44
  Join Override property for PeopleSoft records 39, 46
  joining sources for PeopleSoft records 39
  linking TO_EFFDT port for PeopleSoft records 50
  PeopleSoft, configuring 58
  PeopleSoft, default join 39
  PeopleSoft, mapping parameters and variables 38
  PeopleSoft, overview 36
  PeopleSoft, XLATTABLE SQL override 62
  Select Distinct option, for a PeopleSoft record 49
  sorting ports for PeopleSoft records 49
architecture
  PeopleSoft 12

B

branch nodes
  PeopleSoft, importing levels 28
branches
  PeopleSoft, description 15
business names
  displaying business names in PeopleSoft 22

C

column names
  joining PeopleSoft records 40
connect string
  to import PeopleSoft metadata 9
connectivity
  PeopleSoft requirements 9
  PeopleSoft, overview 12

D

data extraction
  default query for PeopleSoft records 38
  Extract Override property for PeopleSoft records 47
  horizontal PeopleSoft trees 18
  PeopleSoft import permissions 11
  PeopleSoft records 14
  PeopleSoft, username 10
  vertical PeopleSoft trees 19
data extraction *(continued)*
  XLATTABLE 60, 61, 63
database table name
  for PeopleSoft records 23, 37
datatypes
  horizontal PeopleSoft trees 28
  PeopleSoft native datatypes 71
  PowerExchange for PeopleSoft 70
  unsupported in PeopleSoft 71
dates
  filtering PeopleSoft records 14
default join
  Application Source Qualifier for PeopleSoft sources 39
  PeopleSoft records 40
  PeopleSoft tree and detail record 41
  PeopleSoft tree and non-detail record 43
  PeopleSoft, effective dates 40
  PeopleSoft, effective sequences 40
default query
  Application Source Qualifier, for PeopleSoft records 38
  editing for PeopleSoft records 39
  extract current rows from PeopleSoft records 50
  generating for PeopleSoft records 39
  PeopleSoft, overriding for XLATTABLE 62
  PeopleSoft, specifying current date 50
detail ranges
  PeopleSoft join behavior 41
  PeopleSoft, importing 28
detail records
  importing from PeopleSoft 30
  importing name with PeopleSoft tree 26
  joining with PeopleSoft trees 41
  PeopleSoft, description 15
  relationship to PeopleSoft detail tree 17
detail trees
  flattening 19
  joining with PeopleSoft records 41
  PeopleSoft, description 15
  PeopleSoft, static detail ranges 17
  PeopleSoft, strict-level example 17
  vertically flattening 21
distinct values
  selecting from a PeopleSoft record 49

E

EFFDT
  description 60
Effective Date Join Order
  PeopleSoft, description 51
  PeopleSoft, examples 53
effective dates
  default join for PeopleSoft records 40
  filtering PeopleSoft records 14
  joining PeopleSoft records 51

effective dates *(continued)*
PeopleSoft TO_EFFDT port 14
PeopleSoft TO_EFFDT port, adding 50
PeopleSoft trees, importing 29, 31
PeopleSoft, importing 26
PeopleSoft, in XLATTABLE 60
selecting current rows from PeopleSoft records 50
trees 28
effective sequences
default join for PeopleSoft records 40
filtering PeopleSoft records 14
EFFSEQ
PeopleSoft, description 14
Extract Current Rows
PeopleSoft, description 51
PeopleSoft, overview 50
Extract Date
mapping parameters and variables in PeopleSoft 50
PeopleSoft, description 51
PeopleSoft, mapping parameters and variables 38
PeopleSoft, overview 50
extract override
PeopleSoft, Application Source Qualifier 36
PeopleSoft, creating 48
PeopleSoft, validating 48
Extract Override (property)
Application Source Qualifier connected to PeopleSoft records 47

F

FIELDNAME
in XLATTABLE lookup SQL override 65
in XLATTABLE source filter 62
PeopleSoft, description 60
FIELDVALUE
in XLATTABLE join condition 62
in XLATTABLE lookup condition 64
PeopleSoft, description 60
filtering
PeopleSoft, record business names 31
PeopleSoft, record names 31
PeopleSoft, tree names 31
PeopleSoft, XLATTABLE data 62
XLATTABLE data 65
filters
PeopleSoft, validating 45
flattening
PeopleSoft trees 41

J

join override
PeopleSoft, creating 46
PeopleSoft, validating 46
Join Override (property)
PeopleSoft, Application Source Qualifier 36, 39, 46
Joiner transformation
PeopleSoft, joining sources 41
joining
detail PeopleSoft trees and non-detail records 43
PeopleSoft records and detail trees 41
PeopleSoft records and trees 41
related PeopleSoft records 39, 40
joins
PeopleSoft, creating key relationships for 14

K

keys
joining PeopleSoft records 40
PeopleSoft, creating for joins 14, 25
PeopleSoft, handling 14
PeopleSoft, importing 14
PeopleSoft, in records 25
PeopleSoft, primary 14
PeopleSoft, source definitions 14

L

language codes
in XLATTABLE 60
PeopleSoft, overview 72
language table name
PeopleSoft, for records 23
LANGUAGE_CD
in XLATTABLE lookup SQL override 65
in XLATTABLE source filter 62
PeopleSoft, description 60
levels
PeopleSoft, description 15
PeopleSoft, skipping 16, 18, 19
location information
PeopleSoft, Lookup transformation 67
PeopleSoft, session property sheet 67
lookup SQL override
PeopleSoft, validating 66
Lookup transformation
creating for XLATTABLE data 64
lookup SQL override for XLATTABLE data 65
PeopleSoft, adding owner name 65
PeopleSoft, cache 66
PeopleSoft, configuring database connection 67
PeopleSoft, configuring session 67
PeopleSoft, location information 67
PeopleSoft, lookup condition 64
XLATTABLE data 60
loose-level trees
description 16
PeopleSoft source definition, creating 29
winter tree example 16

M

mapping parameters
in Application Source Qualifier for PeopleSoft 38
mapping variables
PeopleSoft, in Application Source Qualifier 38
metadata extensions
PeopleSoft, using 23
metadata tables
PeopleSoft, import permissions 10
PeopleSoft, username and permissions for import 9
PeopleSoft, username for 10

N

nodes
PeopleSoft, missing 16, 18, 19
non-detail records
joining with PeopleSoft trees 43

O

ODBC data sources
  PeopleSoft, configuring 11
  PeopleSoft, to import metadata 9
ORDER BY
  extract query for PeopleSoft records 49
override
  extract for PeopleSoft records 47
  PeopleSoft, default join 46
  PeopleSoft, XLATTABLE SQL override 62
  XLATTABLE lookup SQL override 65
owner name
  in PeopleSoft sessions 69
  PeopleSoft table 10
  XLATTABLE table 63, 65

P

panels
  PeopleSoft, records 24
partitioning
  PeopleSoft, sessions 69
PeopleSoft
  architecture 12
  connectivity 12
  metadata table permissions 10
  security 13
  source table permissions 11
permissions
  PeopleSoft, to extract source data 9
  PeopleSoft, to import metadata tables 9
  XLATTABLE table 63
physical table name
  for PeopleSoft records 23
primary keys
  PeopleSoft, importing 14

R

records
  filtering PeopleSoft record 14
  importing database table name for PeopleSoft 23
  joining with PeopleSoft detail trees 41
  joining with PeopleSoft records 40
  PeopleSoft detail 17
  PeopleSoft, creating key relationships 14
  PeopleSoft, description 14, 22
  PeopleSoft, importing 22–24
  PeopleSoft, importing language table name 23
  PeopleSoft, in panels 24
  PeopleSoft, joining with detail trees 40
  PeopleSoft, joining with records 40
  PeopleSoft, key columns 25
  PeopleSoft, metadata 23
  PeopleSoft, related 25
  PeopleSoft, using database table names in queries 37
  PeopleSoft, viewing 24
  sorting source data from PeopleSoft records 49
root
  PeopleSoft, description 15
root nodes
  PeopleSoft, importing levels 28

S

security
  configuring for PeopleSoft 10
  PeopleSoft metadata tables 10
  PeopleSoft overview 13
  PeopleSoft source tables 11
  XLATTABLE tables 65
Select Distinct (property)
  PeopleSoft, Application Source Qualifier 36
  PeopleSoft, extract query 49
sequences
  filtering PeopleSoft records 14
sessions
  PeopleSoft sources, creating 68
  PeopleSoft, partitioning 69
  PeopleSoft, table name prefix 69
Set Control Value
  PeopleSoft, importing 26
SetID
  PeopleSoft, importing 26
  trees 28
sorted ports
  PeopleSoft, Application Source Qualifier 36
  PeopleSoft, overview 49
  PeopleSoft, procedure 49
Sorter transformation
  PeopleSoft data, sorting 49
source definitions
  default query for PeopleSoft records 38
  displaying business names in PeopleSoft 22
  Extract Override property for PeopleSoft records 47
  filtering data from PeopleSoft sources 44
  horizontal PeopleSoft trees 18
  PeopleSoft records 14
  PeopleSoft, editing 34
  PeopleSoft, extract override 36
  PeopleSoft, filtering data 36
  PeopleSoft, for XLATTABLE data 62
  PeopleSoft, importing 31
  PeopleSoft, joining 36, 40, 46
  PeopleSoft, selecting distinct values 36
  PeopleSoft, sorting source data 36
  PeopleSoft, XLATTABLE data 60
  selecting distinct values from a PeopleSoft record 49
  sorting source data from PeopleSoft records 49
  vertical PeopleSoft trees 19
source filters
  Application Source Qualifier for PeopleSoft records 44
  PeopleSoft, Application Source Qualifier 36
  PeopleSoft, creating 45
  PeopleSoft, syntax 44
  PeopleSoft, validating 45
source tables
  PeopleSoft, username and permissions for import 9
  permissions to secure PeopleSoft data 11
SQL override
  PeopleSoft, XLATTABLE 62
SQL table
  PeopleSoft, description 14, 23
  PeopleSoft, key columns 25
  PeopleSoft, key handling 14
SQL view
  PeopleSoft, description 14, 23
  PeopleSoft, key columns 25
  PeopleSoft, key handling 14
strict-level trees
  description 16

strict-level trees *(continued)*
example 16
missing nodes 16
PeopleSoft, detail tree example 17
structures
trees 17
summary trees
description 18
flattening 19
PeopleSoft, description 15
vertically flattening 21
syntax
PeopleSoft, filters 44

T

table names
PeopleSoft, language 23
PeopleSoft, physical 23
TO_EFFDT port
description 14
linking 50
trees
denormalizing 18
description 22
detail data from PeopleSoft trees 30
detail PeopleSoft records 30
effective dates 28, 29, 31
flattening 18, 19, 41
flattening horizontally 18
flattening vertically 19
importing 22
importing attributes 29
importing horizontal 28
joining with PeopleSoft records 41, 43
joining with records 40
loose-level 16
mapping parameters and variables 38
missing nodes 18, 19
overview 15, 26
PeopleSoft detail records 17
PeopleSoft metadata 26
PeopleSoft, creating source definition 33
trees *(continued)*
PeopleSoft, description 15
PeopleSoft, importing attributes 33
PeopleSoft, supported 15
SetID 28, 29, 31
summary 17
vertical flattening 33

U

username
PeopleSoft database 10, 11

W

winter trees
description 15, 18
flattening 18
supported 15
vertically flattening 20

X

XLATLONGNAME
description 60
XLATSHORTNAME
description 60
XLATTABLE data
description 60
extracting 60
looking up 60, 63
overview 60
sourcing 61
XLATTABLE table
as lookup table 64
configuring database connections 63
owner name 63, 65
PeopleSoft source definition, importing 62, 63
permissions 63
source filter, entering 62