PeopleSoft®

PeopleTools 8.42 Integration Tools PeopleSoft Integration Tools and Utilities

November 2002

PeopleTools 8.42
Integration Tools
PeopleSoft Integration Tools and Utilities
SKU TOOLS842INT-B 1102

PeopleBooks Contributors: Teams from PeopleSoft Product Documentation and Development.
Copyright 1988-2002 PeopleSoft, Inc. All rights reserved.

Printed in the United States.

All material contained in this documentation is proprietary and confidential to PeopleSoft, Inc. ("PeopleSoft"), protected by copyright laws and subject to the nondisclosure provisions of the applicable PeopleSoft agreement. No part of this documentation may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, including, but not limited to, electronic, graphic, mechanical, photocopying, recording, or otherwise without the prior written permission of PeopleSoft.

This documentation is subject to change without notice, and PeopleSoft does not warrant that the material contained in this documentation is free of errors. Any errors found in this document should be reported to PeopleSoft in writing.

The copyrighted software that accompanies this document is licensed for use only in strict accordance with the applicable license agreement which should be read carefully as it governs the terms of use of the software and this document, including the disclosure thereof.

PeopleSoft, PeopleTools, PS/nVision, PeopleCode, PeopleBooks, PeopleTalk, and Vantive are registered trademarks, and Pure Internet Architecture, Intelligent Context Manager, and The Real-Time Enterprise are trademarks of PeopleSoft, Inc. All other company and product names may be trademarks of their respective owners. The information contained herein is subject to change without notice.

Open Source Disclosure

This product includes software developed by the Apache Software Foundation (http://www.apache.org/). Copyright (c) 1999-2000 The Apache Software Foundation. All rights reserved. THIS SOFTWARE IS PROVIDED “AS IS” AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

PeopleSoft takes no responsibility for its use or distribution of any open source or shareware software or documentation and disclaims any and all liability or damages resulting from use of said software or documentation.

Contents

About This PeopleBook

PeopleBooks provide you with the information that you need to implement and use PeopleSoft applications.

This preface discusses:

- PeopleSoft application prerequisites.
- PeopleSoft application fundamentals.
- Related documentation.
- Typographical elements and visual cues.
- Comments and suggestions.
- Common elements in PeopleBooks.

Note. PeopleBooks document only page elements that require additional explanation. If a page element is not documented with the process or task in which it is used, then either it requires no additional explanation or it is documented with common elements for the section, chapter, PeopleBook, or product line. Elements that are common to all PeopleSoft applications are defined in this preface.

PeopleSoft Application Prerequisites

To benefit fully from the information that is covered in these books, you should have a basic understanding of how to use PeopleSoft applications.

See *Using PeopleSoft Applications.*

You might also want to complete at least one PeopleSoft introductory training course.

You should be familiar with navigating the system and adding, updating, and deleting information by using PeopleSoft windows, menus, and pages. You should also be comfortable using the World Wide Web and the Microsoft Windows or Windows NT graphical user interface.

These books do not review navigation and other basics. They present the information that you need to use the system and implement your PeopleSoft applications most effectively.

PeopleSoft Application Fundamentals

Each application PeopleBook provides implementation and processing information for your PeopleSoft database. However, additional, essential information describing the setup and design of your system appears in a companion volume of documentation called the application fundamentals PeopleBook. Each PeopleSoft product line has its own version of this documentation.

The application fundamentals PeopleBook consists of important topics that apply to many or all PeopleSoft applications across a product line. Whether you are implementing a single application, some combination of applications within the product line, or the entire product line, you should be familiar with the contents of this central PeopleBook. It is the starting point for fundamentals, such as setting up control tables and administering security.

Related Documentation

This section discusses how to:

- Obtain documentation updates.
- Order printed documentation.

Obtaining Documentation Updates

You can find updates and additional documentation for this release, as well as previous releases, on the PeopleSoft Customer Connection Website. Through the Documentation section of PeopleSoft Customer Connection, you can download files to add to your PeopleBook Library. You'll find a variety of useful and timely materials, including updates to the full PeopleSoft documentation that is delivered on your PeopleBooks CD-ROM.

Important! Before you upgrade, you must check PeopleSoft Customer Connection for updates to the upgrade instructions. PeopleSoft continually posts updates as the upgrade process is refined.

See Also

PeopleSoft Customer Connection Website, http://www.peoplesoft.com/corp/en/login.asp

Ordering Printed Documentation

You can order printed, bound volumes of the complete PeopleSoft documentation that is delivered on your PeopleBooks CD-ROM. PeopleSoft makes printed documentation available for each major release shortly after the software is shipped. Customers and partners can order printed PeopleSoft documentation by using any of these methods:

- Web
- Telephone
- Email

Web

From the Documentation section of the PeopleSoft Customer Connection Website, access the PeopleSoft Press Website under the Ordering PeopleBooks topic. The PeopleSoft Press Website is a joint venture between PeopleSoft and Consolidated Publications Incorporated (CPI), the book print vendor. Use a credit card, money order, cashier's check, or purchase order to place your order.

Telephone

Contact CPI at 800 888 3559.

Email

Send email to CPI at psoftpress@cc.larwood.com.

See Also

PeopleSoft Customer Connection Website, http://www.peoplesoft.com/corp/en/login.asp

Typographical Conventions and Visual Cues

This section discusses:

- Typographical conventions.
- Visual cues.

Typographical Conventions

The following table contains the typographical conventions that are used in PeopleBooks:

Typographical Convention or Visual Cue	Description
Bold	Indicates PeopleCode function names, method names, language constructs, and PeopleCode reserved words that must be included literally in the function call.
Italics	Indicates field values, emphasis, and PeopleSoft or other book-length publication titles. In PeopleCode syntax, italic items are placeholders for arguments that your program must supply. We also use italics when we refer to words as words or letters as letters, as in the following: Enter the number 0, not the letter *O*.
KEY+KEY	Indicates a key combination action. For example, a plus sign (+) between keys means that you must hold down the first key while you press the second key. For ALT+W, hold down the ALT key while you press W.
`Monospace font`	Indicates a PeopleCode program or other code example.
(quotation marks)	Indicate chapter titles in cross-references and words that are used differently from their intended meanings.

Typographical Convention or Visual Cue	Description
. . . (ellipses)	Indicate that the preceding item or series can be repeated any number of times in PeopleCode syntax.
{ } (curly braces)	Indicate a choice between two options in PeopleCode syntax. Options are separated by a pipe (|).
[ ] (square brackets)	Indicate optional items in PeopleCode syntax.
& (ampersand)	When placed before a parameter in PeopleCode syntax, an ampersand indicates that the parameter is an already instantiated object. Ampersands also precede all PeopleCode variables.
(ISO)	Information that applies to a specific country, to the U.S. federal government, or to the education and government market, is preceded by a three-letter code in parentheses. The code for the U.S. federal government is USF; the code for education and government is E&G, and the country codes from the International Standards Organization are used for specific countries. Here is an example: (GER) If you're administering German employees, German law requires you to indicate special nationality and citizenship information for German workers using nationality codes established by the German DEUEV Directive.
Cross-references	PeopleBooks provide cross-references either below the heading See Also or on a separate line preceded by the word *See.* Cross-references lead to other documentation that is pertinent to the immediately preceding documentation.

Visual Cues

PeopleBooks contain the following visual cues.

Notes

Notes indicate information that you should pay particular attention to as you work with the PeopleSoft system.

Note. Example of a note.

A note that is preceded by *Important!* is crucial and includes information that concerns what you must do for the system to function properly.

Important! Example of an important note.

Warnings

Warnings indicate crucial configuration considerations. Pay close attention to warning messages.

Warning! Example of a warning.

Comments and Suggestions

Your comments are important to us. We encourage you to tell us what you like, or what you would like to see changed about PeopleBooks and other PeopleSoft reference and training materials. Please send your suggestions to:

PeopleSoft Product Documentation Manager PeopleSoft, Inc. 4460 Hacienda Drive Pleasanton, CA 94588

Or send email comments to doc@peoplesoft.com.

While we cannot guarantee to answer every email message, we will pay careful attention to your comments and suggestions.

Common Elements in These PeopleBooks

As of Date	The last date for which a report or process includes data.
Business Unit	An ID that represents a high-level organization of business information. You can use a business unit to define regional or departmental units within a larger organization.
Description	Enter up to 30 characters of text.
Effective Date	The date on which a table row becomes effective; the date that an action begins. For example, to close out a ledger on June 30, the effective date for the ledger closing would be July 1. This date also determines when you can view and change the information. Pages or panels and batch processes that use the information use the current row.
Once, Always, and **Don't Run**	Select Once to run the request the next time the batch process runs. After the batch process runs, the process frequency is automatically set to Don't Run. Select Always to run the request every time the batch process runs. Select Don't Run to ignore the request when the batch process runs.

Report Manager	Click to access the Report List page, where you can view report content, check the status of a report, and see content detail messages (which show you a description of the report and the distribution list).
Process Monitor	Click to access the Process List page, where you can view the status of submitted process requests.
Run	Click to access the Process Scheduler request page, where you can specify the location where a process or job runs and the process output format.
Request ID	An ID that represents a set of selection criteria for a report or process.
User ID	An ID that represents the person who generates a transaction.
SetID	An ID that represents a set of control table information, or TableSets. TableSets enable you to share control table information and processing options among business units. The goal is to minimize redundant data and system maintenance tasks. When you assign a setID to a record group in a business unit, you indicate that all of the tables in the record group are shared between that business unit and any other business unit that also assigns that setID to that record group. For example, you can define a group of common job codes that are shared between several business units. Each business unit that shares the job codes is assigned the same setID for that record group.
Short Description	Enter up to 15 characters of text.

See Also

Using PeopleSoft Applications

PeopleSoft Process Scheduler

PeopleSoft Integration Tools and Utilities Preface

This preface provides an overview of the PeopleSoft Integration Tools and Utilities PeopleBook.

PeopleSoft Integration Tools and Utilities

In addition to PeopleSoft Integration Broker, Component Interfaces and Business Interlinks, PeopleTools includes file layout technology and the PeopleSoft API Repository. This PeopleBook describes how you can use these products to facilitate the implementation of your integration strategy.

CHAPTER 1

File Layouts and Data Interchange

This chapter provides an overview of file layouts and discusses how to:

- Construct file layouts.
- Customize file layouts.
- Perform data interchange.
- Produce a sample file layout.

Understanding File Layouts

This section provides an overview of file layouts and discusses:

- Identifying fields in a flat file.
- Fixed positional (FIXED) format files.
- Comma separated value (CSV) format files.
- XML tagged (XML) format files.

File Layout Overview

A *file layout* is a definition that represents the data structure of a flat (text) file to be processed. When reading from or writing to a flat file, your PeopleCode can use a file layout as a template to identify and correctly process the file's records and fields. File layouts work with hierarchical and non-hierarchical data, and can handle files that combine data records with non-data (audit or control) records.

To access data in a file, you don't have to create a file layout. PeopleTools supports reading and writing to plain text files, as well as to files that have a format based on a file layout.

- If the file is a plain text file, data is read or written using text strings.
- If the file is based on a file layout, you can use text strings, rowset or record objects.

Using a file layout greatly simplifies reading, writing and manipulating hierarchical transaction data with PeopleCode. It can facilitate transactions between your PeopleSoft application and a third party system when the third party system doesn't support Integration Broker or component interfaces.

Applying File Layouts to Data Interchange

In addition to manipulating transaction data, you can employ file layouts to move data between your PeopleSoft database and external systems (data interchange), using flat files as the transmission medium. File layouts enable you to:

- Export hierarchical PeopleSoft data to several flat file formats.
- Map incoming hierarchical file data directly to PeopleSoft tables.
- Preview and troubleshoot the input data and its formatting before importing it.
- Automatically generate the Application Engine and PeopleCode programs needed to perform data import.
- Use batch processes to perform large volume data import and export.

See Also

Chapter 1, "File Layouts and Data Interchange," Performing Data Interchange, page 16

PeopleCode Reference, "File Class"

Identifying Fields in a Flat File

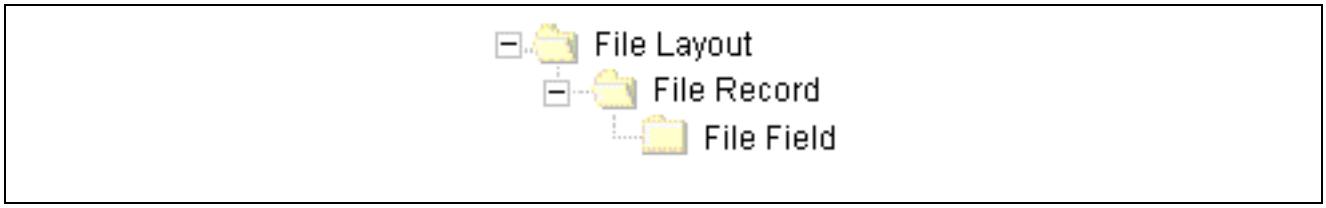
A flat file, in the simplest sense, is a collection of fields in text format. The file must be formatted in a way that enables your PeopleCode to locate each field. PeopleSoft file layouts support three formats:

- *Fixed Position (FIXED):* Each field has a starting position and a length which together specify its location in the file. This is the default format for new file layouts.
- *Comma Separated Values (CSV):* Fields are located in the file by their sequence, separated from each other by commas.
- *XML Tagged (XML):* A field is located not by its position or sequence within a record, but by the named XML tags surrounding it.

To preserve relationships between fields, we need a way to logically group fields into a collection. In relational databases, these collections are records. Each line within a file can be considered a collection of fields.

Note. With some file formats, the logical concept of a record may actually span multiple physical lines, but the concept of collections of fields remains.

The conceptual structure of all three file formats is represented in a file layout as follows:



File layout structure

A file layout is a collection of file records, which in turn are a collection of fields, each of which has a describable location.

Note. To avoid confusion with the standard terms record and field, when working with a file layout we refer to a collection of fields as a file record and to fields as file fields.

Fixed Positional (FIXED) Format Files

This is the most common type of flat file currently processed by PeopleSoft EDI Manager. Almost all EDI type processing uses this file type where each data element is oriented by a fixed, or column dependent, position within the file.

The following table describes the properties of FIXED format files.

Property	Description	EDI Manager Equivalent
File Layout Format	File format (FIXED).	None.
File Record ID	A group of numbers that can be used to identify the file record.	RowID.
ID Start Position	The column starting position of the file record ID.	Treated as a field within each map.
(Record) ID Length	The length of the file record ID.	Treated as a field within each map.
File Record Name	A user specified name for the file record.	PeopleSoft record name.
File Field	A user specified name for the file field.	PeopleSoft record's field name.
(Field) Start Position	The column starting position of the file field.	Starting position.
Field Length	The length of the file field.	Length of field.
Field Format	The formatting options for both inbound and outbound field processing.	Based on field type.

Considerations for FIXED Format

You should be aware of the following when working with FIXED format files:

- Be careful when you change the length or starting position of any file fields, or if you insert a new file field between two existing ones. It's possible to overlay fields. Results are unpredictable.
- When you insert a record into a file layout, fields of type Long are converted to type Character, with a length of 0. You must set a field length greater than 0 before you can save the file layout.

Comma Separated Value (CSV) Format Files

In this type of file, each data element is surrounded with a separator, a delimiter or both. File Record IDs can be used to determine which table data is moved to or from, however, in most cases this type of file contains homogenous records.

In the following example, the qualifier is double quotes (“) and the delimiter is a comma (,).

```
"NAME","ADDRESS","PHONE"
```

File layout definitions store the file record ID (when used) and the relative sequence number of the field. (In the text example above, “PHONE” is sequence number 3).

The following table describes the properties of CSV format files.

Property	Description	EDI Manager Equivalent
File Format	File format (CSV).	None.
File Record ID	A group of numbers that can be used to identify the file record.	RowID.
ID Sequence Number	The sequence number of the field that contains the file record ID.	Treated as a field within each map.
Qualifier	The character that surrounds a field to mark its extent. This can be set at the file layout, file record or file field level.	Delimiter.
Delimiter	The character used to separate fields from each other. This can be set at the file layout or file record level.	Separator.
File Record Name	A user specified name for the file record.	None
File Record Field	A user specified name for the file field.	None
Field Format	The formatting options for both inbound and outbound field processing.	Based on Field type

Considerations for CSV Format

You should be aware of the following when working with files of CSV format:

- Both the qualifier and the delimiter accept a blank as a valid value.

- If a field is NULL, you don't have to use qualifiers. In the following example, Field2 is NULL.

```
Field1,,Field3,Field4. . .
```

XML Tagged (XML) Format Files

This type of file contains data represented in a hierarchical or tree type structure. A tag surrounds each data element. A file record tag might group multiple entries.

File layout definitions tie the identifier along with parent and child relationships to the file record and file field.

There is no PeopleSoft EDI Manager equivalent for this format.

The following table describes the properties of XML format files.

Attribute	Description
File Format	The file format (XML).
File Record ID	The tag name representing the file record.
Field Identifier	The tag name representing the file field.
Field Format	The formatting options for both inbound and outbound field processing.
File Record Name	A user specified name for the file record.
Field Name	A user specified name for the file field.

Considerations for XML Format

You should be aware of the following when working with files in XML format:

- Your XML input file must contain *at least* the fields that are specified in the file layout definition you're using. If the file is missing any fields, the input rowset won't contain any data.
- If your XML input file contains extra fields that aren't specified in the file layout definition, the ReadRowset method will ignore the extra fields.
- When you insert a record into a file layout, fields of type Long are converted to type Character, with a length of 0. You must set a field length greater than 0 before you can save the file layout.

Constructing File Layouts

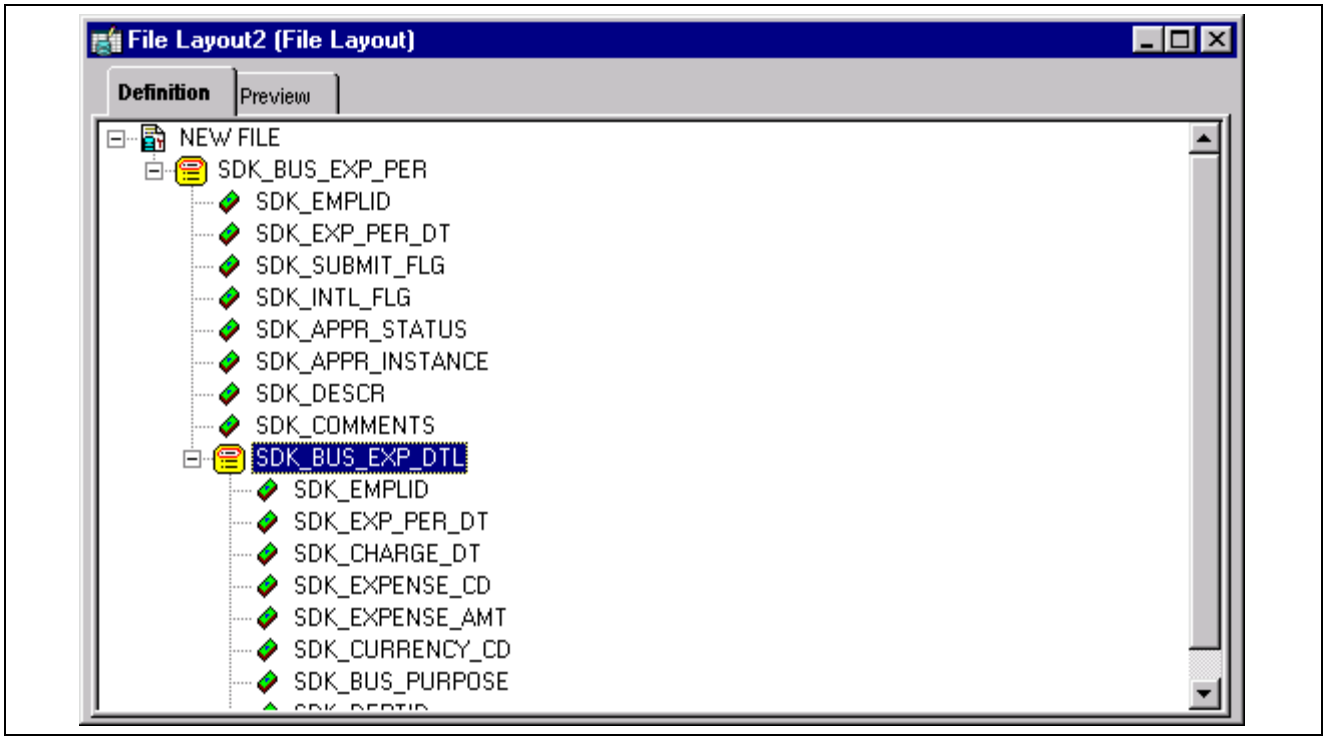
This section discusses how to:

- Create a new file layout definition.

- Add file records and file fields.
- Reorder the file layout nodes.
- Name file layouts, records, and fields.
- Apply field formats.

Creating a New File Layout Definition

In Application Designer, select File, New, File Layout.



File layout with two records in hierarchy

A new file layout contains only one element — the root node, to which file records are attached. The default root node name is *NEW FILE*.

Save the new file layout definition. You'll be prompted to give the file layout a name, which will also be the name of the root node.

Note. The default file layout format is FIXED.

Adding File Records and File Fields

You can add file records and file fields to your file layout in two ways:

- You can base them on existing record and field definitions in your database.
- You can construct them directly in your file layout, without reference to any database records or fields. A file record you construct this way is added as a *segment,* but the result is an ordinary file record.

You can use both methods in any combination when creating file records and file fields in a file layout definition. Each file record and file field is generically referred to as a *node*.

Important! Regardless of the method you use, file records and file fields exist only as definitions within a file layout, and have no connection with any database record or field. Even with file records based on records in your database, a change to the database record definition will *not* be reflected in the file layout.

When to Use a Segment Instead of a Record

Suppose that in the file provided to you, some of the file records contain new data and need to be inserted, while others contain data that updates existing data. You could add a segment with a single field (like AUDIT_ACTION) that indicates whether the file record is new or changed. When you process the file, you can use PeopleCode to look at this field and, based on its value, do the appropriate action.

Another example: suppose you wanted to include two fields from the PERSONAL_DATA table in your file, but not all the other fields. You have two choices: insert the PERSONAL_DATA table and manually delete all the unwanted fields, or insert a segment, name it PERSONAL_DATA, then insert the two fields you want.

Segments in Data Interchange

If you're creating a file layout for the purpose of data interchange, you can use segments, but each file record must correspond to a record with the same name in your PeopleSoft database, and its file fields must have the same names and data formats as the database record's fields. The file record may contain a subset of the fields in the corresponding database record. It may also contain a subset or a superset of the fields provided in the corresponding file data.

Adding File Records

To add a file record to your file layout definition, use one of the following methods:

- Drag and drop a database record.

  Drag a record definition from the Project window into the file layout's Definition window, dropping it on the root node or any existing file record or file field. The new file record appears at the same level as the node you dropped it on, following all the other file records at that level. All of the record's constituent fields are inserted as well.

- Insert a database record.

  With the root node or a file record highlighted, select Insert, Record. When you select a record, the new file record appears following the highlighted file record, at the same level. All of the record's constituent fields are automatically inserted as well.

- Insert a segment.

  With the root node or a file record highlighted, select Insert, Segment, and enter a file record name. When you click OK, the new file record appears following the highlighted file record, at the same level.

- Insert a child segment.

  With a file record highlighted, select Insert, ChildSegment, and enter a file record name. When you click OK, the new file record appears one level below the highlighted file record, before any others at that level.

Note. When you add a file record at the root level, it appears immediately below the root node, before all the other file records.

Adding File Fields

To add a file field to your file layout definition, use one of the following methods:

- Drag and drop a database field.

  Drag a field definition from the Project window into the file layout's Definition window, dropping it on any existing file record or file field. Confirm the field name or enter a different one, and click OK. The new file field appears following the node you dropped it on.

- Insert a database field.

  With a file record or file field highlighted, select Insert, Database Field. Confirm the field name or enter a different one, and click OK. The new file field appears following the highlighted node.

- Insert a file field.

  With a file record or file field highlighted, select Insert, FileField, and enter a file field name. When you click OK, the new file field appears following the highlighted node.

Note. Each file field must have a unique name within its parent file record, but file fields in different file records can have the same name.

Reordering File Layout Nodes

The file layout definition provides a set of directional arrow buttons in the toolbar, which you can use to reposition any file record within the hierarchy of the file layout, or any file field within its parent file record.

The up and down arrows don't change the level of the selected item, just its order among other items at that level. The right and left arrows move the selected item lower and higher in the file layout hierarchy.

Note. When you reposition a file record in the file layout, its child records are also repositioned, and their child records, and so on.

Naming File Layouts, Records and Fields

File layout names can be 30 characters in length, and file record and file field names can be 15 characters in length, and all should follow PeopleSoft naming standards.

Each file record within a file layout must have a unique name, but one file record can have the same name as the file layout. Each file field within a given file record must have a unique name, but file fields in different file records can have the same name.

Using WriteRecord, ReadRowset and WriteRowset

If you use the WriteRecord, ReadRowset or WriteRowset file layout methods for writing to or reading from records, the application record and the file record *must have the same name*, and the application record fields and the file fields *must have the same names*. These methods only write to like-named records and like-named fields within a given record. If you rename a record or a field after you use it to create a file layout definition, you will have to rename your file record or file field to the exact same name.

In a file layout definition containing more than one record, records and fields that aren't like-named are ignored. Like-named records don't have to contain all the same fields, and like-named fields don't have to be the same length. PeopleSoft recommends that like-named fields be of the same type.

See Also

PeopleSoft Application Designer, "Creating Record Definitions," Naming Record Definitions

Applying Field Formats

You must take several issues into consideration when specifying field formats.

Outputting Numbers to FIXED Files

When you write numeric data to a FIXED format flat file, all numbers are written right justified in the file field. Numbers with decimal places specified will be written with zeros padding the unused decimal places.

For example, a sequence of records with numbers of varying precision are written this way:

```
001    53.2700BUY
002  2174.0933SELL
003   108.0000SELL
```

Date, Time, and Datetime Field Considerations

In accordance with ISO 8601 standards, the field lengths for Date, Time and Datetime fields are fixed in the file layout, regardless of file format:

- Date fields have a fixed length of 10.
- Time fields have a fixed length of 20.
- Datetime fields have a fixed length of 31.

Considerations for Using Dates With the ReadRowset Method

Single digits in dates in the form MMDDYY or MMDDYYYY must be padded with zeros. That is, if the date in your data is February 3, 2002, the form must be:

- 02/03/2002
- 02/03/02

The following is *not* valid: 2/3/02.

See Also

PeopleCode Reference, "File Class," File Layout Error Processing

Customizing File Layouts

This section discusses how to:

- Specify file layout properties.
- Specify file record properties.
- Specify file field properties.

Note. Some properties are only available for a specific file layout format. For example, a file definition tag is only available for a file with XML specified as the file layout format. When a property is only available for a particular format, that is noted in parentheses after the name of the property (such as File Definition Tag (XML)).

Specifying File Layout Properties

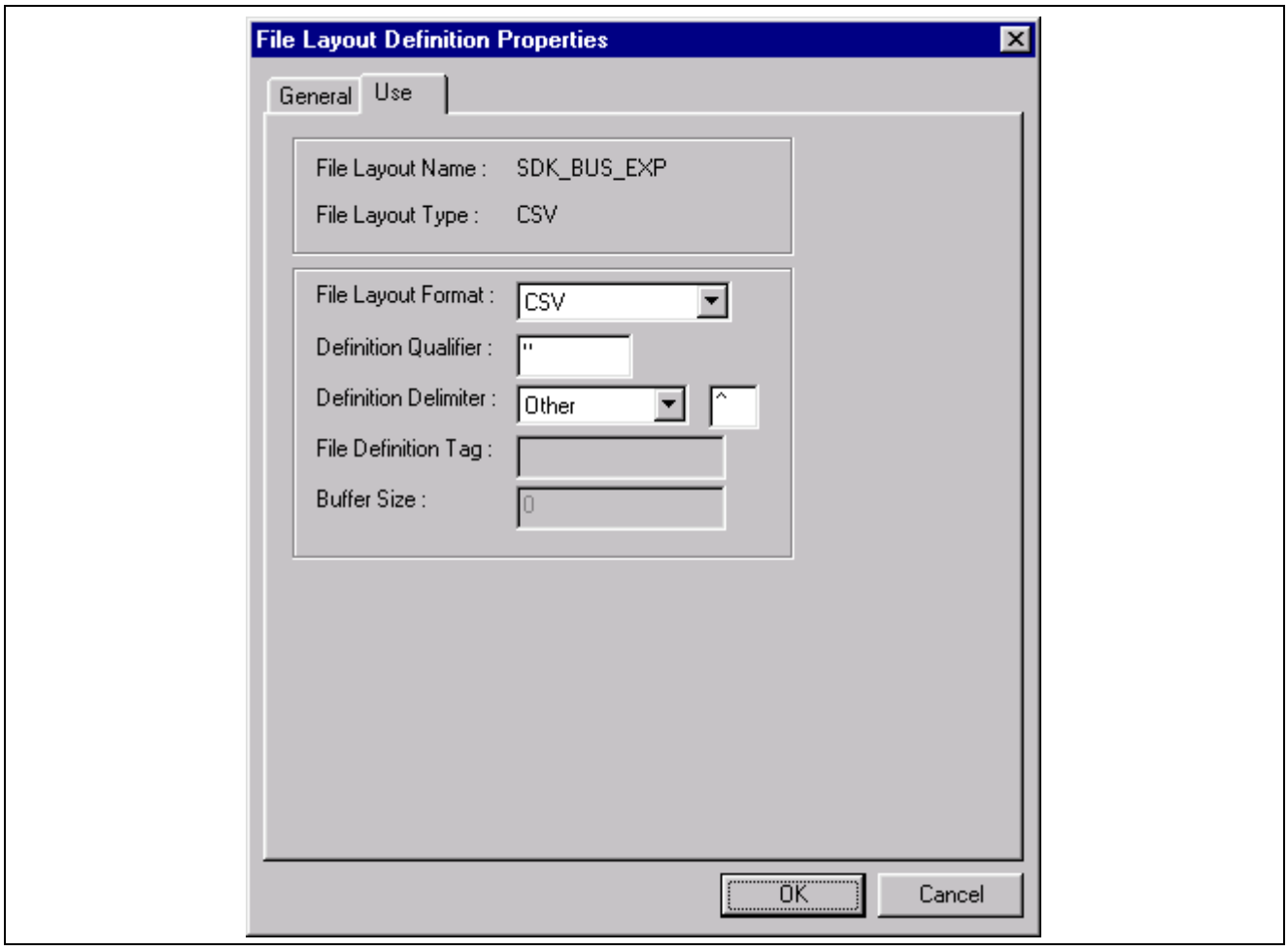
The File Layout Definition Properties dialog box contains all information stored at the file layout (root) level.

You can access the dialog box using one of the following methods:

- Use File, Object Properties.
- Press ALT+ENTER.
- Double-click the topmost (root) node of a file layout definition.
- Right-click an open file layout, then select Data Object Properties.

The General tab of the dialog box contains description information for the file layout.

The Use tab contains specific information for the file layout.



File Layout Definition Properties: Use tab

File Layout Format The type of file layout. Valid values are *FIXED*, *CSV*, or *XML*.

Definition Qualifier (CSV) Select a qualifier to surround each field in the file record in this layout This value can be overridden at the file record and file field levels.

Definition Delimiter (CSV) Select the default delimiter between each field in the file record in this layout. This value can be overridden at the file record level. The default value is *Comma*. If you specify *Other*, a blank field appears so you can enter a delimiter.

File Definition Tag (XML) Enter the XML tag name associated with this layout (or transaction). This tag can be 30 characters in length. This tag must be unique in the file layout.

Buffer Size (XML) Displays the size of the input buffer used at runtime.

Note. This value shouldn't be edited directly.

See Also

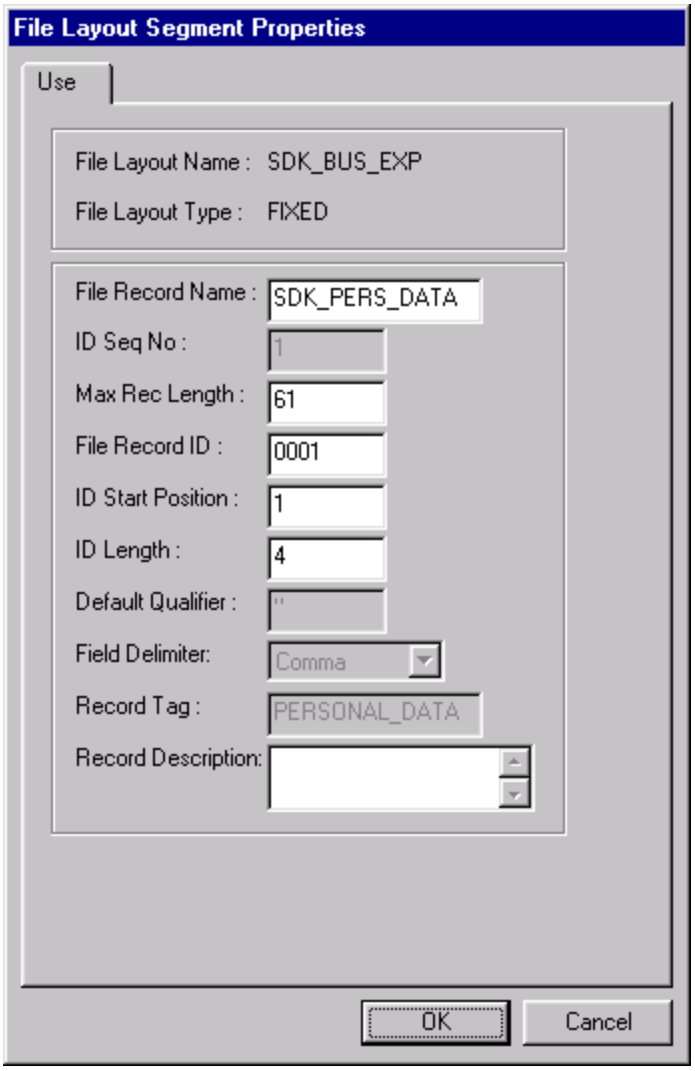
Chapter 1, "File Layouts and Data Interchange," Understanding File Layouts, page 1

Specifying File Record Properties

The File Layout Segment Properties dialog box contains information stored at the file record level.

You can access the dialog box using one of the following methods:

- Double-click the file record node.
- Select the file record node, right-click, then select Selected Node Properties.



File Layout Segment Properties dialog box

File Record Name Enter a file record name associated with this file record. This name is used when accessing the file record from PeopleCode. Every file record in a file layout must have a unique name.

ID Seq No. (CSV) Enter a sequence number for the field that contains the file record ID.

Max Rec Length Displays the default maximum length of the combined field sizes of the record. This value is automatically updated.

Warning! Any inbound or outbound data is truncated beyond this value.

File Record ID Enter a number to uniquely identify the file record in the file layout. You can use this number in processing the file. This number is automatically written to the file if you use the WriteRecord or WriteRowset methods and the file type is FIXED or CSV.

ID Start Position (FIXED) Enter the column or starting position in the file record where the file record ID starts.

ID Length (FIXED, CSV) Displays the length of the file record ID. This number is automatically generated when you enter the File Record ID.

Note. This value shouldn't be edited directly.

Default Qualifier (CSV) Enter a qualifier used for the file record ID and the default for fields when no field qualifier is specified. This value overrides the definition qualifier specified in the File Layout Definition Properties dialog box. When you first create a file layout, this property is blank.

Field Delimiter (CSV) Enter a delimiter used for all fields in the file record. This overwrites the definition delimiter specified on the File Layout Definition Properties dialog box.

Record Tag (XML) Enter an XML tag name for this file record. The default value is the file record name.

Note. Although each record name in a file layout must be unique, record tags do not have to be unique.

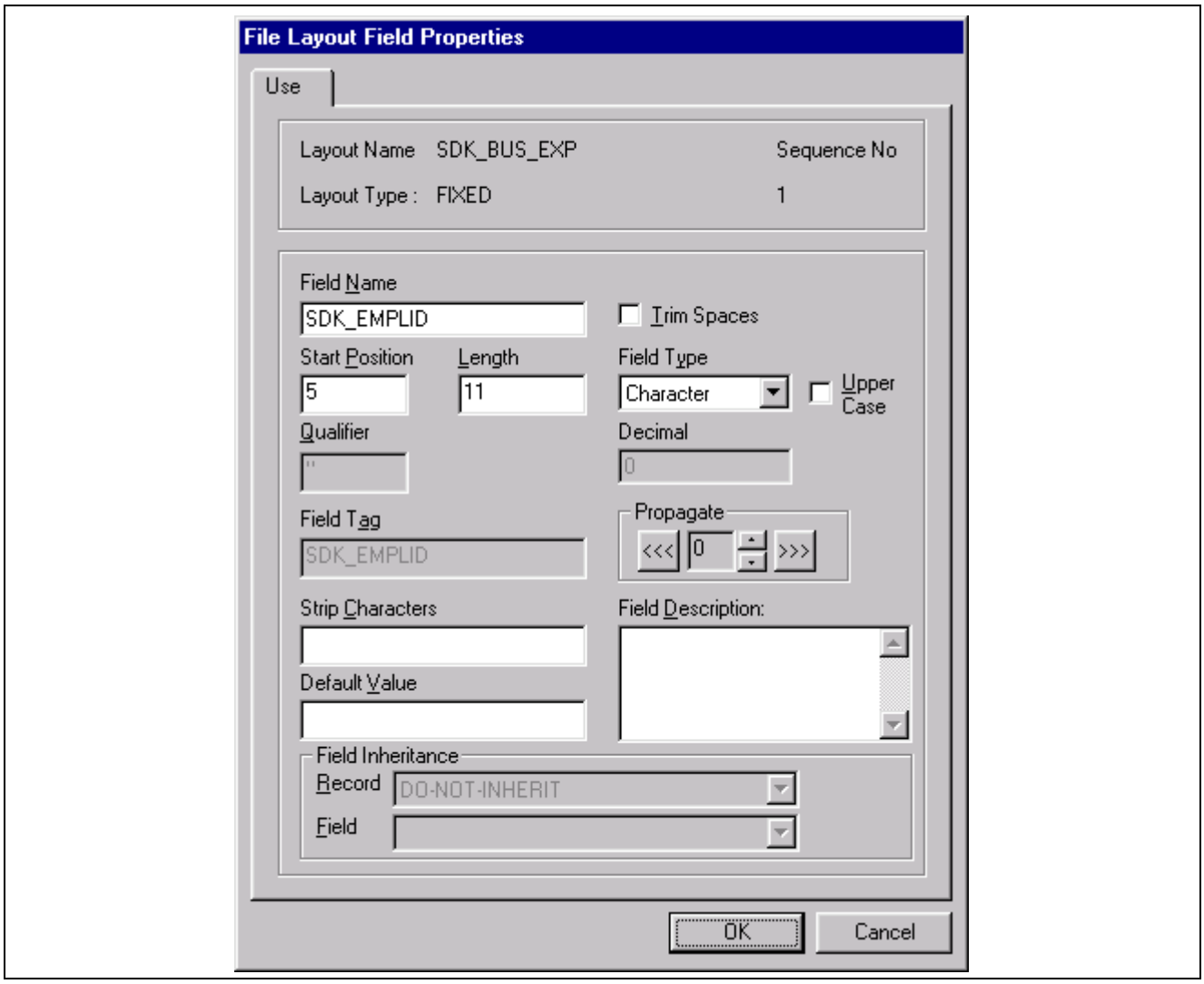
Record Description Enter a description of the record, for documentation purposes only.

Specifying File Field Properties

The File Layout Field Properties dialog box contains information stored at the file field level.

You can access the dialog box using one of the following methods:

- Double-click the file field node.
- Select the file field node, right-click, then select Selected Node Properties.



File Layout Field Properties dialog box

Most individual properties are usable by all field types. However, some are specific to a particular field type — for example, the *UpperCase* checkbox is only applicable for character fields, while the *Date Separator* field is only applicable for date fields, and so on. The dialog box pictured here shows the properties for a character type of field. However, the following description goes through all possible properties.

Field Name Enter the name associated with this file field. This name is used when accessing the file field from PeopleCode. Every field within a file record must have a unique name; however, two different file records can contain the same file field.

UpperCase (Char) Select to convert lowercase text to uppercase during inbound processing. Primarily used when customer data may be in lowercase, and PeopleSoft requires the data to be in uppercase.

Field Type Select the data type of the file field.

Date Format (Date) Select a date format, such as MMDDYY, DDYYMM, and so on.

Date Separator (Date) Enter a character used to separate date values. The default value is /.

Decimal Position Enter the number of decimal positions (to the right) of the decimal point. This property is only valid for fields defined as Number or Signed Number.

Note. You're only allowed 31 characters plus a decimal point.

Field Length Enter the maximum number of characters of this field.

Note. You're only allowed 32 character precision for number and signed number fields, that is, a total of 32 characters both to the right and left of the decimal. Other fields, such as character fields, can be longer.

Note. You can't set the field length for fields of type Date, Time, and Datetime. These field lengths are automatically set to the ISO standards for such fields.

Start Position (FIXED) Enter the starting position (column) of the field within the file record.

Important! If you specify a start position for a field that overwrites a previous field, no data is written to the file. Use Propagate to change the start positions for your file fields.

Propagate (FIXED) If a field position or length is changed, enter an amount here to increment (positive number) or decrement (negative number) the current field and all fields before it (<<<) or after it (>>>).

Field Qualifier (CSV) Enter the qualifier for the field, that is, the character that surrounds this field, separating it from other fields. Specifying this value overwrites the value specified in the file layout properties and file record properties.

Field Tag (XML) Enter an XML tag name to be used around the field. The default value is the name of the field.

Note. Although each field name in a file record must be unique, each field tag does not have to be unique.

Strip Characters Specify any characters to be removed from the input buffer. You use this to preprocess input strings. For example, if a field in your input file contains hyphens, but you want to remove the hyphens prior to processing the field, you could enter a hyphen here, and it would be stripped out while being read. You can specify more than one character to be stripped out. Be sure to not separate the strip characters. For example, the following strips out all semi-colons and hyphens:

```
;-
```

The following strips out all semi-colons, hyphens, and spaces:

```
; -
```

Trim Spaces Select to remove the leading and trailing spaces from the input string but leave spaces within the string intact.

This is different from the Strip Characters field, which will remove all spaces from the entire input field if you specify a space.

Field Description Enter a description of the field for documentation purposes.

Field Inheritance

Optionally select a parent file record and field, from which the current field's value is to be inherited. If you're writing to a file, this means the value will only be written in the parent file record, not the child (inheriting) file record (that is, the value won't be written more than once to a file.) If no value is present in the parent field, the default value specified here will be used.

For example, the following file sample shows both the EMPLID (8113) and EFFDT (08/06/1999) written only once to a file, though these fields are repeated in the third file record (with File Record ID 102.)

```
100 8113 Frumman,Wolfgang
101      08/06/1999     000001   219 Going to London office
102                 100 000015 I 08/06/1999
102                 200 000030 I 08/06/1999
102                 300 000009 I 08/06/1999
102                 400 000001 I 08/06/1999
102                 500 000011 I 08/06/1999
```

See Also

PeopleSoft Application Designer, "Creating Field Definitions"

Performing Data Interchange

This section provides an overview of the import process and discusses how to:

- Preview the input data.
- Generate and run the import program.
- Export data.

Understanding the Import Process

To help you troubleshoot and import flat file data, the file layout definition provides a data preview page. It can also generate an Application Engine program with associated PeopleCode necessary to import the data.

Data Import Activities

Importing data using a file layout requires the following sequence of activities:

1. Provide the import data in a properly formatted flat file. Each record in the file must correspond to a record with the same name in your PeopleSoft database, and its fields must have the same names and data formats as the database record's fields. Each record in the file must end with a newline character.
2. Create a file layout definition to match the record and field structure of your data. Insert the appropriate record definitions into your file layout, then reposition the file records and file fields to match the record and field positions in your file.
3. Preview and troubleshoot the input data format and content.
4. Generate the data import Application Engine program and PeopleCode.

5. Run the Application Engine program to import the data.

Including and Excluding Fields

The fields in your data file's records can be a subset of the database record's fields — define your file layout with only the file fields you expect to receive.

The fields in your data file's records can be a superset of the database record's fields; you must define your file layout to suppress or ignore the extra fields. For FIXED files, don't define a field at the corresponding position in your file layout definition. For CSV files, the file layout must have the same number of fields in each record as there are in the corresponding file record — for each field you don't want to import, define a field in the file layout at that position that *doesn't* correspond to any field in that database record. For XML files, any extra fields are ignored automatically.

Your data file can contain a subset or a superset of the records defined in the file layout. Only a file record with a matching file record ID in the file layout is imported.

Record Hierarchy

In theory, you can ignore rowset hierarchy when importing file data, because the PeopleSoft database stores each record independently of the others, and rowsets aren't used in the import process. However, many records are designed with hierarchical dependencies in mind. The input file might omit inherited field values or order the data records in a way that reflects such dependencies.

If your input file omits inherited field values, make sure the inheriting fields' records in the file layout are children of the ones from which they inherit their values, and make the appropriate Field Inheritance settings.

If the records to be imported contain key fields that reflect a rowset hierarchy, they might be in an order in the file that also reflects the hierarchy. Make sure your file layout reflects that hierarchy as well.

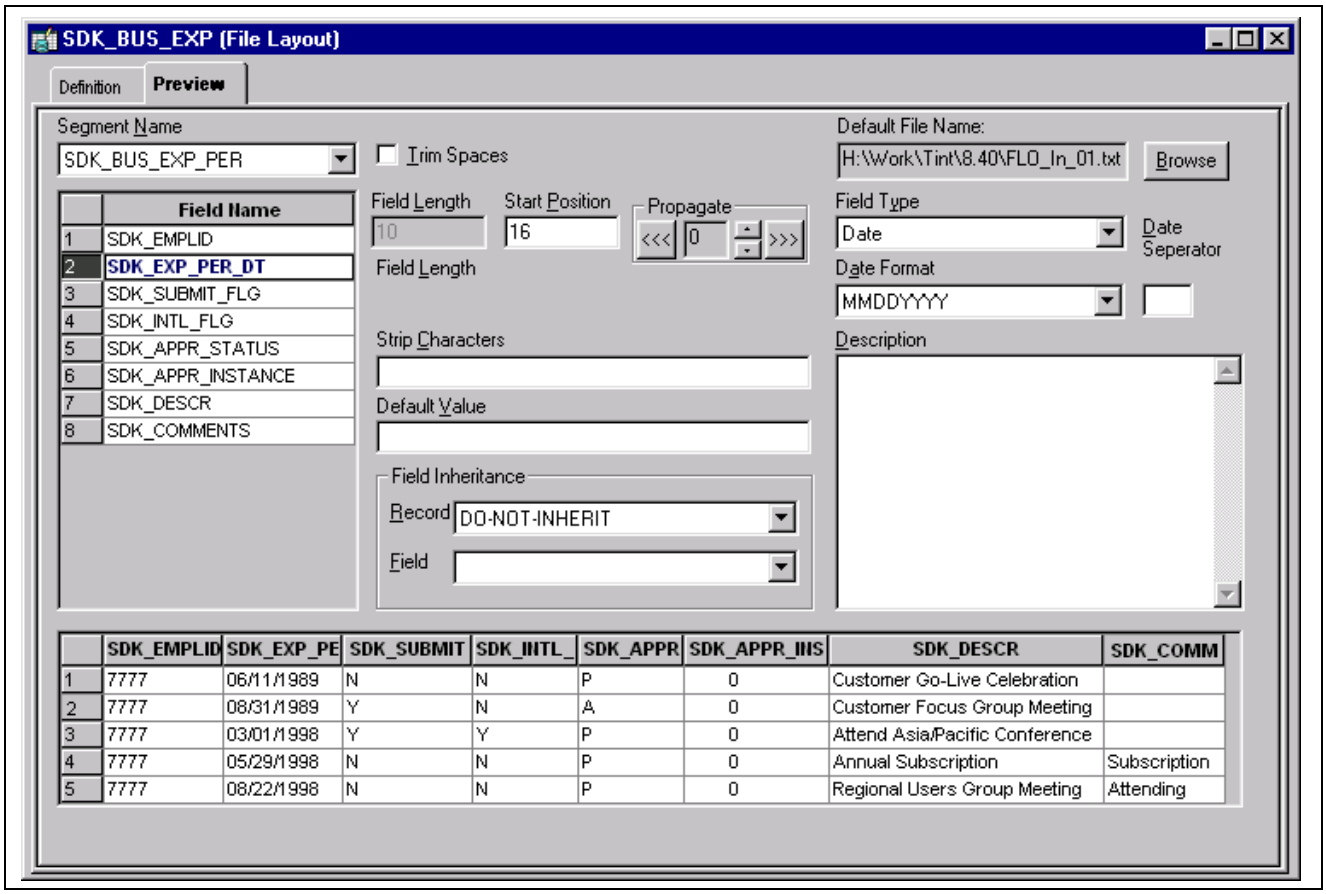
Important! Your completed file layout must have exactly one file record at the root level; all other file records must be below that level.

See Also

Chapter 1, "File Layouts and Data Interchange," Reordering File Layout Nodes, page 8

Previewing Input Data

In Application Designer, select File, New, File Layout, Preview.



File layout preview

Initially no information is displayed on the preview tab.

Note. With an XML file the data can't be previewed, although you can still generate and run an import program.

To preview the input data:

1. Click Browse and select your import file.

   The name of the import file appears in the Import File text box.

2. Select a file layout segment from the Segment dropdown list.

   The file layout fields corresponding to the selected segment appear in the list below the segment name. This list comes from the file layout definition, and does not depend on you selecting a valid import file.

3. Click the Refresh button on the toolbar to refresh the preview data.

   At the bottom of the window, a preview of the first five rows of the selected segment in the selected import file appears in the grid.

   You can also select View, Automatically Read Preview Data. While this menu item is selected, the file layout refreshes the preview automatically; when it's cleared, you need to click the Refresh Preview Data button to see the effects of changes you make to the property settings or data.

Note. Preview file data is available only for Fixed and CSV file layouts. The preview data only appears if you've selected an import file, if the file format matches the format specified for the file layout definition, and if it contains a file record ID that matches the file record ID of the selected segment.

4. Select one of the fields on the Field Name list.

   The properties of the field you selected appear in the appropriate display fields. These properties are actually the field's file layout field properties; you can change all field property settings from this view.

Note. Only the field properties appropriate to the file layout format you specified in the File Layout Definition Properties dialog box are visible in this view; for example, the Start Position and Field Length fields are available only for a FIXED format file layout.

Examining the Input Data for Errors

The following table shows some examples of input data errors.

Symptom	Possible Reason	Solution
The preview grid doesn't appear.	The input file's format doesn't match the file layout format you specified in the File Layout Definition Properties dialog box.	Change the file layout format to match the input file.
	A CSV file doesn't use the definition qualifier you specified in the File Layout Definition Properties dialog box.	Change the definition qualifier to match the one used in the input file.
Only the first column of the preview grid is populated.	A CSV file doesn't use the definition delimiter you specified in the File Layout Definition Properties dialog box.	Change the definition delimiter to match the one used in the input file.
The preview grid appears for some records, but not for others.	The file's record IDs for the missing records don't match their file record IDs specified in the File Layout Segment Properties dialog box.	Specify file record IDs that match the input file records.
Data for a field appears truncated in the preview grid.	With a FIXED file, the field length you specified in the File Layout Field Properties dialog box is too short to accommodate the field data.	Increase the field length to accommodate the input data.
A field appears to start in the middle of the data.	With a FIXED file, the start position you specified in the File Layout Field Properties dialog box is too great to include the start of the field data.	Decrease the start position and adjust the field length to match the input data start position and length.

Generating and Running the Import Program

This part of the process imports the file data.

To generate and run the import program:

1. After you preview your file layout and examined the input data, click the AE button on the toolbar.

   This will generate the Application Engine import program with its associated import PeopleCode.

2. Enter a name for the Application Engine program and click OK.

   The program is automatically saved and is ready to run as soon as its definition appears in Application Designer.

3. Click the Run Program button on the toolbar.

   The file data is imported into your database.

Exporting Data

The method you use to export data from PeopleSoft depends on the target application's requirements. To export data to a flat file, you'll create a file layout definition, then write PeopleCode to transfer the data to a file. The PeopleCode can be launched from Application Engine or any event. It should populate text strings, rowset objects or record objects, and apply the File class WriteRecord or WriteRowset method to transfer the data to the file, using the file layout definition to position the records and fields as required by the target application.

Note. To generate valid XML files, be sure to use the file class Close method when you finish writing to the file.

See Also

PeopleCode Reference, "File Class"

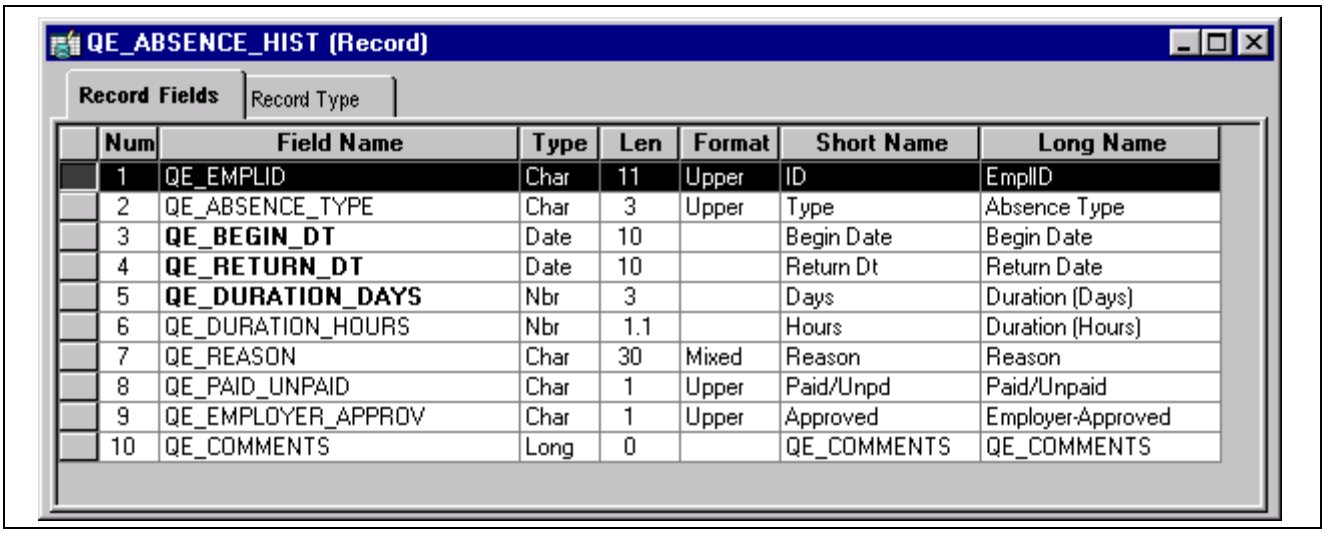
Producing a Sample File Layout

This section describes how to:

- Create the file layout definition.
- Adjust the file layout properties.
- Insert a segment and a file field.

Creating the File Layout Definition

This example illustrates how to create a file layout that could be used with the QE_ABSENCE_HIST record.



QE_ABSENCE_HIST (Record)

Record Fields | Record Type

Num	Field Name	Type	Len	Format	Short Name	Long Name
1	QE_EMPLID	Char	11	Upper	ID	EmplID
2	QE_ABSENCE_TYPE	Char	3	Upper	Type	Absence Type
3	**QE_BEGIN_DT**	Date	10		Begin Date	Begin Date
4	**QE_RETURN_DT**	Date	10		Return Dt	Return Date
5	**QE_DURATION_DAYS**	Nbr	3		Days	Duration (Days)
6	QE_DURATION_HOURS	Nbr	1.1		Hours	Duration (Hours)
7	QE_REASON	Char	30	Mixed	Reason	Reason
8	QE_PAID_UNPAID	Char	1	Upper	Paid/Unpd	Paid/Unpaid
9	QE_EMPLOYER_APPROV	Char	1	Upper	Approved	Employer-Approved
10	QE_COMMENTS	Long	0		QE_COMMENTS	QE_COMMENTS

QE_ABSENCE_HIST record definition

For simplicity, let's say each row in our Fixed format data file has the following structure:

```
888 A
000 8001 VAC  1981-09-12 1981-09-26 14  .0                    P Y
888 A
000 8001 VAC  1983-03-02 1983-03-07 5   .0                    P Y
888 A
000 8001 VAC  1983-08-26 1983-09-10 13  .0                    P Y
888 A
000 8516 MAT  1986-06-06 1986-08-01 56  .0                    P Y
888 C
000 8516 SCK  1988-08-06 1988-08-07 1   .0                    P Y
888 A
000 8516 VAC  1987-07-14 1987-07-28 14  .0                    P Y
888 A
000 8553 JUR  1990-12-12 1990-12-17 5   .0  Local Jury Duty    P N
888 A
000 8553 MAT  1992-02-20 1992-10-01 224 .0  Maternity Leave    U N
888 A
000 8553 MAT  1994-08-19 1995-03-01 194 .0  Maternity          U Y
888 A
000 8553 PER  1993-04-15 1993-04-19 4   .0                    U N Personal
Day required
888 C
000 8553 SCK  1987-01-28 1987-01-30 2   .0  Hong Kong Flu      P N
888 A
000 8553 SCK  1988-08-02 1988-08-03 1   .0  Sick               P N
888 A
000 8553 SCK  1995-09-12 1995-09-13 1   .0                    P N
888 C
000 G001 MAT  1991-07-02 1991-09-28 88  .0  3-month Maternity  P Y Maternity will
be paid as 80% of Claudia's current salary.
```

000 is the file record ID for ABSENCE_HIST, and each field appears in the same order as in the ABSENCE_HIST database record. 888 is the file record ID for an extra segment called CHANGE_ACTION, containing an AUDIT_ACTION field with the following meanings:

- A: Row inserted.
- C: Row updated, but no key fields changed.

Note. The end of file (EOF) character must be on a separate line and not on a line containing data for any incoming file, regardless of file type. Each data line needs to be terminated with an end of line (EOL) character, which is different than an EOF.

To create a file layout definition:

1. Use the QE_ABSENCE_HIST record definition as a template for the file layout.

   Create a new file layout, then drag the QE_ABSENCE_HIST record into the open file layout.

2. Save the file layout.

   Save it with a name of ABS_HIST.

   The name of the first node changes from NEW FILE to ABS_HIST.

Adjusting File Layout Properties

In the file layout definition, open the file layout, record, or field properties as appropriate for each step in the following procedure.

To adjust layout, record, and field properties:

1. Change the file layout properties.

   Double-click the topmost node in the file layout, ABS_HIST, to display the file layout properties. Fill in a short and long description of the file layout you're creating. For this example, we're creating a FIXED file layout, so you don't need to make any changes on the Use tab.

2. Change the file record properties.

   Double-click the QE_ABSENCE_HIST file record to display its properties. Enter a record ID of 000, and a starting position of 1. The ID length is automatically set.

   When you click OK, you'll get a message asking if you want to increment the start positions for all fields.

   Click Yes. This will automatically increment the start position numbers for every field to take the length of the file Record ID you just added into account. If you don't click yes, you'll have to manually increment the start position for all your fields.

   We've just created the file record ID for the QE_ABSENCE_HIST record:

```
000 8001       VAC  1981-09-12 1981-09-26 14   .0                    P Y
000 8001       VAC  1983-03-02 1983-03-07 5    .0                    P Y
```

3. Change the file field properties.

   When a record definition is used as a template for a file layout, the default starting position for each field is based on the order it appears in the record as well as its length.

   Double-click the QE_EMPLID file field to display its properties.

The start position is automatically incremented to 4 (since the file record ID is three characters long). However, in the example there's an extra space between the end of the file record ID and the first field. Therefore, you need to change the start position of this field, and all the fields after this field. This increments the starting position of this field and all fields following this field by 1.

To do this:

- Click the up arrow under Propagate to change that number from 0 to 1.
- Click the button with the arrows pointing right (>>>).

4. Adjust other fields (optional).

The last field (QE_COMMENTS) has a length of 0. This is because it's based on a field of type Long.

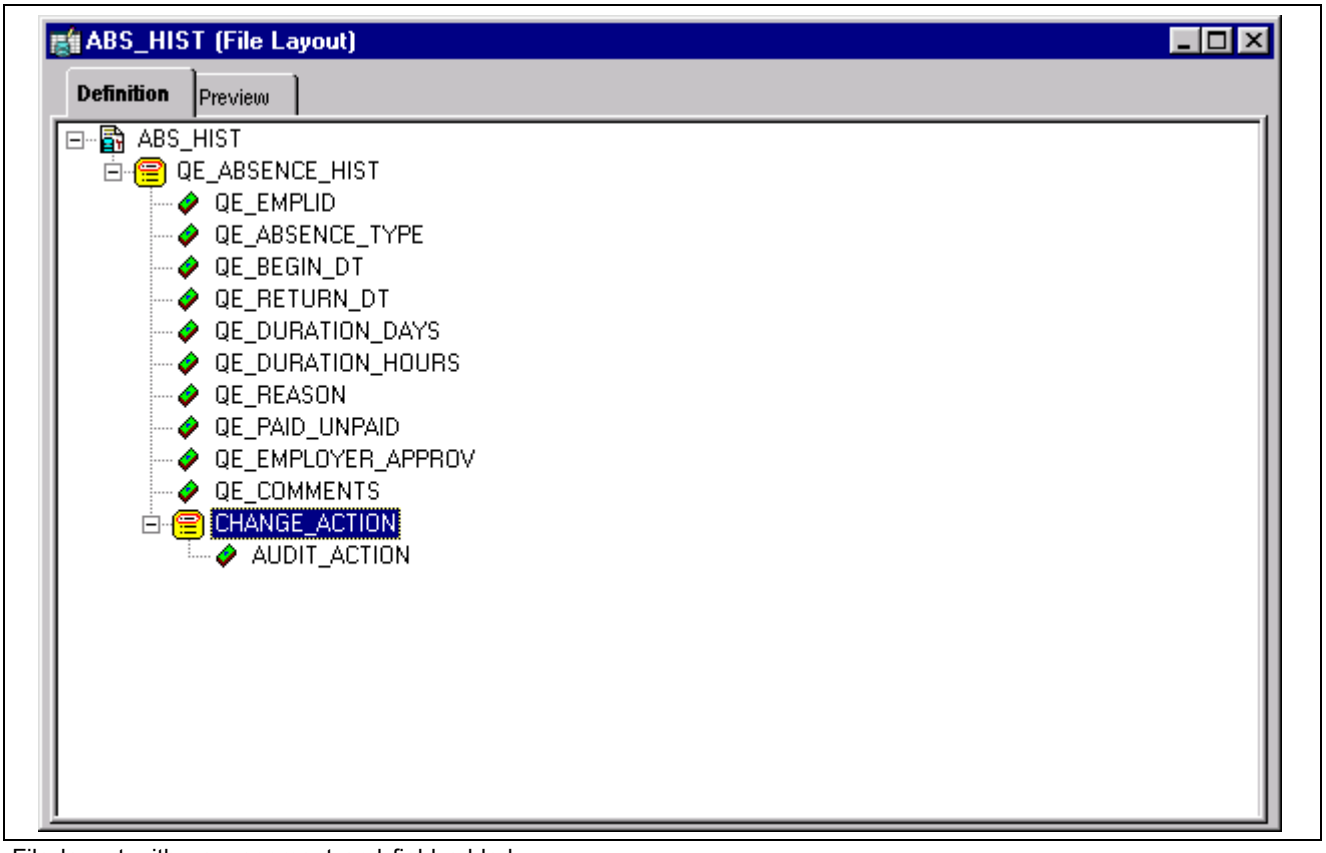
Note. When a Long field is inserted into a file layout, it's converted to a Character field with a length of 0.

Because the format for this file layout is FIXED, you have to change the field length of that file field so it's long enough to accommodate the data you expect from the file.

You don't have to propagate this change because this is the last field in the record.

Inserting a Segment and Field

Open the file layout definition based on the ABS_HIST record.



File layout with new segment and field added

To insert a segment and a field in that segment:

1. Insert a segment.

   We want to insert a segment that is a sibling, (that is, at the same level), as the QE_ABSENCE_HIST record. Insert a segment by selecting Insert Segment from the pop-up menu, or by selecting Insert, Segment. The Insert New Segment dialog box is displayed.

   This dialog box is identical to the File Layout Segment Properties dialog box. Fill in the File Record Name, File Record ID, and ID Start Position fields. When you click OK, the segment is inserted.

2. Insert a file field.

   Insert a file field by selecting Insert FileField from the pop-up menu. The Insert New Field dialog box appears. It's identical to the File Layout Field Properties dialog box. Fill in the Field Name, Start Position, and Field Length fields, then click OK.

   Note. The start position isn't automatically set when you add a file field to the file record; for this example the AUDIT_ACTION field requires a start position of 5.

   ---

3. Save your work.

   Be sure to save the changes you've made to your file layout by going to File, Save, or clicking the save icon in the toolbar.

   Now that you've created and saved a file layout, you must use PeopleCode to access the data. file layouts rely solely on PeopleCode as the engine behind the actual data access and movement.

See Also

PeopleCode Reference, "File Class"

CHAPTER 2

The PeopleSoft API Repository

This chapter provides an overview of the PeopleSoft API Repository and describes:

- Repository properties.
- Bindings collection properties.
- Bindings collection methods.
- Bindings properties.
- Bindings methods.
- Namespaces collection properties.
- Namespaces collection methods.
- Namespaces properties.
- Namespaces methods.
- ClassInfo collection properties.
- ClassInfo collection methods.
- ClassInfo properties.
- MethodInfo collection properties.
- MethodInfo collection methods.
- MethodInfo properties.
- PropertyInfo collection properties.
- PropertyInfo collection methods.
- PropertyInfo properties.
- Summary of repository methods and properties.

Understanding the PeopleSoft API Repository

This section provides an overview of the PeopleSoft API repository and how to access the repository using PeopleCode and Visual Basic.

PeopleSoft API Repository Overview

The PeopleSoft API Repository allows PeopleCode and third-party integrators to discover the internally available classes, methods, and properties provided by PeopleSoft for integration. The repository is useful to third-party integrators who integrate in a generic fashion: middleware providers, testing tool providers, and automated documentation providers.

The PeopleSoft API Repository is not a necessary interface for integrators who integrate at the business rule level, such as integration with an expense report, and so on. Those integrators should use PeopleSoft Component Interfaces or PeopleSoft Business Interlinks.

The repository describes available PeopleSoft APIs and provides mechanisms to determine the classes available in the API, the properties of each class, the methods of a class (along with the required parameters), and information concerning which *group* a class belongs to (known as a namespace).

The process of determining information about the API is known as *discovery*. Third-party integrators use information found through discovery to drive generic integration tools.

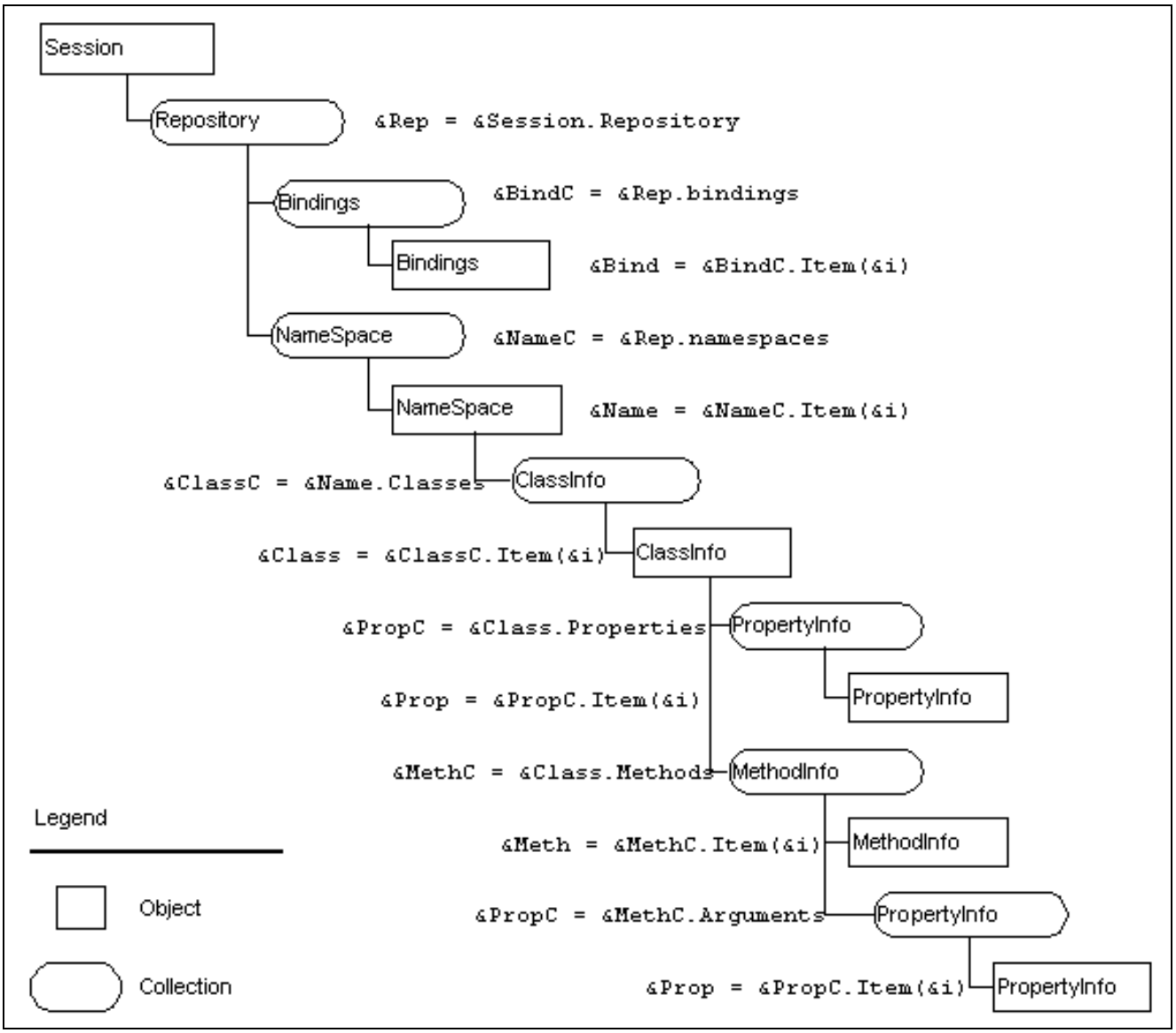
The repository is divided into namespaces. Each namespace contains a collection of related classes. Example namespaces include "PeopleSoft," "ComponentInterface," "Trees," and "BusinessInterlinks".

A *class* defines a related set of methods and properties. Using the repository, you can determine the methods and properties that are available and can be used on any object returned by a call to the PeopleSoft API. An *instance* of a class is known as an *object*.

A *property* is a data item of an object that has both a name and type (string, number, and Boolean, and so on, are examples of types). Some properties are used for inputting data to a class, some are used for getting data from a class, and some are used for both. Whether a property is used for input or output or both is known as *usage*.

A *method* is a function you can call on an object. Methods have a name and a return type (string, number, Boolean, and so on). Methods also have a collection of arguments that must be set prior to invoking the method. Methods arguments have identical attributes to properties.

The following diagram shows the different types of objects and collections instantiated from the repository:



Repository model

Accessing the Repository Using PeopleCode

The following example gets information for the class ABS_HIST from the Namespace component interface and writes it to the file BC.TXT

The following is the complete code sample, followed by the flat file. The next section presents steps that explain each line.

```
Local ApiObject &MYSESSION;
Local ApiObject &MYCI;
Local string &OutTEXT;
Local File &MYFILE;

&MYSESSION = GetSession();
&MYSESSION.Connect(1, "EXISTING", "", "", 0);
```

```
&MYFILE = GetFile("CI.txt", "A");

&NAMESPACES = &MYSESSION.Repository.Namespaces;
&NAMESPACE = &NAMESPACES.ItemByName("CompIntfc");

&OutTEXT = "Namespace = " | &NAMESPACE.Name;
&MYFILE.WriteLine(&OutTEXT);

&CLASSES = &NAMESPACE.classes;
&CLASS = &CLASSES.ItemByName("ABS_HIST");

&OutTEXT = "  Class: " | &CLASS.Name;
&MYFILE.WriteLine(&OutTEXT);

&OutTEXT = "     Methods";
&MYFILE.WriteLine(&OutTEXT);

&METHODS = &CLASS.methods;
For &K = 0 To &METHODS.Count - 1
   &METHOD = &METHODS.item(&K);
   &OutTEXT = "             " | &METHOD.name | ":  " | &METHOD.Type

   &MYFILE.WriteLine(&OutTEXT);
   &ARGUMENTS = &METHOD.arguments;
   For &M = 0 To &ARGUMENTS.count - 1
      &ARGUMENT = &ARGUMENTS.item(&M);
      &OutTEXT = "           " | &ARGUMENT.name | ": " | &ARGUMENT.type;
      &MYFILE.WriteLine(&OutTEXT);
   End-For;
End-For;

&OutTEXT = "     Properties";
&MYFILE.WriteLine(&OutTEXT);

&PROPERTIES = &CLASS.properties;
For &I = 0 To &PROPERTIES.count - 1
   &PROPERTY = &PROPERTIES.item(&I);
   &OutTEXT = "              " | &PROPERTY.name | ": " | &PROPERTY.type;
   &MYFILE.WriteLine(&OutTEXT);
End-For;
&MYFILE.Close();
```

The above code produces the following flat file:

```
Namespace = CompIntfc
  Class: ABS_HIST
     Methods
             Get:  Boolean
             Save:  Boolean
```

```
            Cancel:  Boolean
            Find:  ABS_HIST
            GetPropertyByName:  Variant
          Name: String
            SetPropertyByName:  Number
          Name: String
          Value: Variant
            GetPropertyInfoByName:  CompIntfcPropertyInfo
          Name: String
      Properties
           EMPLID: String
           LAST_NAME_SRCH: String
           NAME: String
           ABSENCE_HIST: ABS_HIST_ABSENCE_HISTCollection
           interactiveMode: Boolean
           getHistoryItems: Boolean
           componentName: String
           compIntfcName: String
           stopOnFirstError: Boolean
           propertyInfoCollection: CompIntfcPropertyInfoCollection
           createKeyInfoCollection: CompIntfcPropertyInfoCollection
           getKeyInfoCollection: CompIntfcPropertyInfoCollection
           findKeyInfoCollection: CompIntfcPropertyInfoCollection
```

Understanding the PeopleCode Example

The following procedure goes through the PeopleCode example line by line.

To retrieve information from the API Repository:

1. Get a session object.

   Before you can access the PeopleSoft API Repository, you have to get a session object. The session controls access to PeopleSoft, provides error tracing, allows you to set the runtime environment, and so on.

   ```
   &MYSESSION = GetSession();
   &MYSESSION.Connect(1, "EXISTING", "", "", 0);
   ```

2. Open the file.

   As this text will be written to a flat file, the next step is to open the file. If the file is already created, the new text is appended to the end of it. If the file hasn't been created, the GetFile built-in function creates the file.

   ```
   &MYFILE = GetFile("CI.txt", "A");
   ```

3. Get the namespace you want.

   Use the Namespaces property on the repository object to get a collection of available namespaces. We want to discover information about a component interface, so we specify CompIntfc in the ItemByName method to get that namespace. With ItemByName, you must specify a namespace that already exists. You'll receive a runtime error if you specify one that doesn't exist.

   ```
   &NAMESPACES = &MYSESSION.Repository.Namespaces;
   &NAMESPACE = &NAMESPACES.ItemByName("CompIntfc");
   ```

4. Write the text to the file.

   Because all the information discovered is being written to a file, the next step is to write text to the file. This code writes the string *Namespace,* followed by the name of the namespace, to the file.

```
&OutTEXT = "Namespace = " | &NAMESPACE.Name;
&MYFILE.WriteLine(&OutTEXT);
```

5. Get the class you want and write text to the file.

   Use the Classes property on the Namespace object to get a collection of all the available classes. We want to discover information about the component interface named ABS_HIST, so we specify that using ItemByName. Then we write that information to the file.

```
&CLASSES = &NAMESPACE.classes;
&CLASS = &CLASSES.ItemByName("ABS_HIST");

&OutTEXT = "  Class: " | &CLASS.Name;
&MYFILE.WriteLine(&OutTEXT);
```

6. Get the methods and arguments, and write the information to the file.

   Use the Methods property on the Class object to get a collection of all the available methods. After you get each method and write the information to the file, loop through and find all of the arguments for the method, then write that information to the file.

```
&OutTEXT = "     Methods";
&MYFILE.WriteLine(&OutTEXT);

&METHODS = &CLASS.methods;
For &K = 0 To &METHODS.Count - 1
   &METHOD = &METHODS.item(&K);
   &OutTEXT = "             " | &METHOD.name | ":  " | &METHOD.Type;
   &MYFILE.WriteLine(&OutTEXT);
   &ARGUMENTS = &METHOD.arguments;
   For &M = 0 To &ARGUMENTS.count - 1
      &ARGUMENT = &ARGUMENTS.item(&M);
      &OutTEXT = "           " | &ARGUMENT.name | ": " | &ARGUMENT.type;
      &MYFILE.WriteLine(&OutTEXT);
   End-For;
End-For;
```

7. Get the properties and write the information to the file.

   Use the Properties property on the Class object to get a collection of all the available properties. Write each property, with its type, to the file. At the end of the program, close the file.

```
&OutTEXT = "     Properties";
&MYFILE.WriteLine(&OutTEXT);

&PROPERTIES = &CLASS.properties;
For &I = 0 To &PROPERTIES.count - 1
   &PROPERTY = &PROPERTIES.item(&I);
   &OutTEXT = "              " | &PROPERTY.name | ": " | &PROPERTY.type;
```

```
   &MYFILE.WriteLine(&OutTEXT);
End-For;
&MYFILE.Close();
```

Accessing the Repository Using Visual Basic

The following example gets information for the class ABS_HIST from the Namespace component interface.

```
Private Sub Command1_Click()
    '************************************************************
    '* TacDemo: Example Repository Usage from Visual Basic
    '*
    '* Copyright (c) 1999 PeopleSoft, Inc.   All rights reserved

    '************************************************************

    ' Declare variables
    Dim oSession As New PeopleSoft_PeopleSoft.Session
    Dim oPSMessages As PSMessageCollection
    Dim oPSMessage As PSMessage

    ' Establish a PeopleSoft Session
    nStatus = oSession.Connect(1, "//PSOFTO070698:9001", "PTDMO", "PTDMO", 0)

    ' Enable error-handler
    On Error GoTo ErrorHandler

    ' Get a Component Interface "shell"
    Dim oNamespaces As NamespaceCollection
    Dim oNamespace As Namespace
    Dim oClasses As ClassInfoCollection
    Dim oClass As ClassInfo
    Dim oMethods As MethodInfoCollection
    Dim oMethod As MethodInfo
    Dim oArguments As PropertyInfoCollection
    Dim oArgument As PropertyInfo
    Dim oProperties As PropertyInfoCollection
    Dim oProperty As PropertyInfo

    Set oNamespaces = oSession.Repository.namespaces
    Set oNamespace = oNamespaces.ItemByName("ComponentInterface")

    Dim outText As String

    outText = "Namespace = " & oNamespace.Name & vbNewLine

    Set oClasses = oNamespace.classes
    Set oClass = oClasses.ItemByName("ABS_HIST")

    outText = outText & "   Class: " & oClass.Name & vbNewLine
```

```
    outText = outText & "       Methods" & vbNewLine

    Set oMethods = oClass.methods
    For k = 0 To oMethods.Count - 1
        Set oMethod = oMethods.Item(k)        outText = outText & "           "
& oMethod.Name & ":  " & oMethod.Type & vbNewLine
        Set oArguments = oMethod.arguments
        For m = 0 To oArguments.Count - 1
            Set oArgument = oArguments.Item(m)
            outText = outText & "                 "
& oArgument.Name & ":  " & oArgument.Type & vbNewLine
        Next
    Next

    outText = outText & "       Properties" & vbNewLine

    Set oProperties = oClass.properties
    For k = 0 To oProperties.Count - 1
        Set oProperty = oProperties.Item(k)
        outText = outText & "            " & oProperty.Name & ":  "
& oProperty.Type & vbNewLine
    Next

 txtResults = outText

' Leave before we encounter the error handler
Exit Sub

ErrorHandler:
    If Err.Number = 1001 Then                ' PeopleSoft Error
        Set oPSMessages = oSession.PSMessages
        If oPSMessages.Count > 0 Then
            For i = 1 To oPSMessages.Count
               Set oPSMessage = oPSMessages.Item(i)
               MsgBox (oPSMessage.Text)
            Next i
            oPSMessages.DeleteAll
        Else
            MsgBox ("PS Api Error.  No additional information
available from Session log")
        End If
    Else                                    ' VB Error
        MsgBox ("VB Error: " & Err.Description)
    End If

End Sub
```

Repository Properties

This section discusses the Repository properties in alphabetical order.

Bindings

Description

The Bindings property returns a reference to a Bindings collection.

This property is read-only.

Namespaces

Description

The Namespaces property returns a reference to a Namespaces collection.

This property is read-only.

Bindings Collection Properties

This section discusses the Bindings collection properties in alphabetical order.

Count

Description

This property returns the number of Bindings Properties objects in the Bindings collection object.

Note. All repository counts begin at zero, not one.

This property is read-only.

Example

```
&COUNT = &BINDINGS.Count;
```

See Also

Chapter 2, "The PeopleSoft API Repository," Bindings Properties, page 34.

Bindings Collection Methods

This section discusses the Bindings collection methods in alphabetical order.

Item

Syntax

Item(*number*)

Description

The Item method returns a Bindings object that exists at the number position in the Bindings collection executing the method

Parameters

Parameter	Description
number	Specify the position number in the collection of the Bindings object that you want returned.

Returns

A reference to a Bindings object or NULL.

Example

```
For &N = 0 to &BINDINGS.Count - 1
   &BINDING = &BINDINGS.Item(&N);
   /* do processing */
End-For;
```

Bindings Properties

This section discusses the Bindings properties in alphabetical order.

Name

Description

This property returns the name of the object as a string.

This property is read-only.

Bindings Methods

This section discusses the Bindings methods in alphabetical order.

Generate

Syntax

```
Generate()
```

Description

This method is a reserved internal function and shouldn't be used at this time.

Namespaces Collection Properties

This section discusses the Namespaces collection properties in alphabetical order.

Count

Description

This property returns the number of Namespaces Properties objects in the Namespaces collection object.

Note. All repository counts begin at zero, not one.

This property is read-only.

Example

```
&COUNT = &NameC.Count;
```

See Also

Chapter 2, "The PeopleSoft API Repository," Namespaces Properties, page 36.

Namespaces Collection Methods

This section discusses the Namespaces collection methods in alphabetical order.

Item

Syntax

```
Item(number)
```

Description

The Item method returns a Namespaces object that exists at the *number* position in the Namespaces collection executing the method.

Parameters

Parameter	Description
number	Specify the position number in the collection of the Namespaces object that you want returned.

Returns

A reference to a Namespaces object or NULL.

Example

```
For &N = 0 to &NAMESPACES.Count - 1
   &NAMESPACE = &NAMESPACES.Item(&N);
   /* do processing */
End-For;
```

ItemByName

Syntax

ItemByName(*name*)

Description

The ItemByName method returns the item specified by *name*. *Name* is not case-sensitive.

Parameters

Parameter	Description
name	Specify the name of the Namespaces object that you want returned. This parameter takes a string value.

Returns

A reference to a Namespaces object or NULL.

Example

```
&NAMESPACE = &NAMESPACES.ItemByName("BusinessComponent");
```

Namespaces Properties

This section discusses the Namespaces properties in alphabetical order.

Classes

Description

This property returns a reference to a ClassInfo collection object.

This property is read-only.

Example

```
&CLASSC = &NAME.Classes;
```

See Also

Chapter 2, "The PeopleSoft API Repository," ClassInfo Collection Properties, page 37.

Name

Description

This property returns the name of the object as a string.

This property is read-only.

Namespaces Methods

This section discusses the Namespaces methods in alphabetical order.

CreateObject

Syntax

CreateObject(*classname*)

Description

This method is a reserved internal function and shouldn't be used at this time.

ClassInfo Collection Properties

This section discusses the ClassInfo collection properties in alphabetical order.

Count

Description

This property returns the number of ClassInfo Properties objects in the ClassInfo collection object.

Note. All repository counts begin at zero, not one.

This property is read-only.

Example

```
&COUNT = &InfoC.Count;
```

See Also

Chapter 2, "The PeopleSoft API Repository," ClassInfo Properties, page 39.

ClassInfo Collection Methods

This section discusses the ClassInfo collection methods in alphabetical order.

Item

Syntax

Item(*number*)

Description

The Item method returns a ClassInfo object that exists at the *number* position in the ClassInfo collection executing the method.

Parameters

Parameter	Description
number	Specify the position number in the collection of the ClassInfo object that you want returned.

Returns

A reference to a ClassInfo object or NULL.

Example

```
For &N = 0 to &CLASSES.Count - 1
   &CLASS = &CLASSES.Item(&N);
   /* do processing */
End-For;
```

ItemByName

Syntax

ItemByName(*name*)

Description

The ItemByName method returns the item specified by *name*. *Name* is not case-sensitive.

Parameters

Parameter	Description
name	Specify the name of the ClassInfo object that you want returned. This parameter takes a string value.

Returns

A reference to a ClassInfo object or NULL.

Example

```
&CLASS = &CLASSES.ItemByName("ABS_HIST");
```

ClassInfo Properties

This section discusses the ClassInfo properties in alphabetical order.

Documentation

Description

This property doesn't actually return all the documentation for the class, just a brief description of the class as a string.

This property is read-only.

Methods

Description

This property returns a reference to a MethodInfo collection object.

This property is read-only.

See Also

Chapter 2, "The PeopleSoft API Repository," MethodInfo Collection Methods, page 40.

Name

Description

This property returns the name of the object as a string.

This property is read-only.

Properties

Description

This property returns a reference to a PropertyInfo collection object.

This property is read-only.

See Also

Chapter 2, "The PeopleSoft API Repository," PropertyInfo Collection Methods, page 43.

MethodInfo Collection Properties

This section discusses the MethodInfo collection properties in alphabetical order.

Count

Description

This property returns the number of MethodInfo Properties objects in the MethodInfo collection object.

Note. All repository counts begin at zero, not one.

This property is read-only.

Example

```
&COUNT = &MethC.Count;
```

See Also

Chapter 2, "The PeopleSoft API Repository," MethodInfo Properties, page 41.

MethodInfo Collection Methods

This section discusses the MethodInfo collection methods in alphabetical order.

Item

Syntax

Item(*number*)

Description

The Item method returns a MethodInfo object that exists at the *number* position in the MethodInfo collection executing the method.

Parameters

Parameter	Description
number	Specify the position number in the collection of the MethodInfo object that you want returned.

Returns

A reference to a MethodInfo object or NULL.

Example

```
For &K = 0 To &METHODS.Count - 1
   &METHOD = &METHODS.item(&K);
   &OutTEXT = "             " | &METHOD.name | ":  " | &METHOD.Type;
   &MYFILE.WriteLine(&OutTEXT);
End-For;
```

ItemByName

Syntax

ItemByName(*name*)

Description

The ItemByName method returns the item specified by *name*. *Name* is not case-sensitive.

Parameters

Parameter	Description
name	Specify the name of the MethodInfo object that you want returned. This parameter takes a string value.

Returns

A reference to a MethodInfo object or NULL.

Example

```
&METHOD = &METHODS.ItemByName("Save");
```

MethodInfo Properties

This section discusses the MethodInfo properties in alphabetical order.

Arguments

Description

This property returns a reference to a PropertyInfo collection object.

This property is read-only.

See Also

Chapter 2, "The PeopleSoft API Repository," PropertyInfo Collection Methods, page 43.

Documentation

Description

This property doesn't actually return all the documentation for the class, just a brief description of the class, as a string.

This property is read-only.

Name

Description

This property returns the name of the object as a string.

This property is read-only.

Type

Description

This property returns the type of the method. Valid values include:

- Bool (Boolean).
- Number.
- Float.
- String.
- Variant.
- Blob (Binary large object).
- Any API class name.

This property is read-only.

PropertyInfo Collection Properties

This section discusses the PropertyInfo collection properties in alphabetical order.

Count

Description

This property returns the number of PropertyInfo Properties objects in the PropertyInfo collection object.

Note. All repository counts begin at zero, not one.

This property is read-only.

Example

```
&COUNT = &PropC.Count;
```

See Also

Chapter 2, "The PeopleSoft API Repository," PropertyInfo Properties, page 44.

PropertyInfo Collection Methods

This section discusses the PropertyInfo collection methods in alphabetical order.

Item

Syntax

Item(*number*)

Description

The Item method returns a PropertyInfo object that exists at the *number* position in the PropertyInfo collection executing the method.

Parameters

Parameter	Description
number	Specify the position number in the collection of the PropertyInfo object that you want returned.

Returns

A reference to a PropertyInfo object or NULL.

Example

```
For &K = 0 To &PROPERTIES.Count - 1
   &PROPERTY = &PROPERTIES.item(&K);
   &OutTEXT = "             " | &PROPERTY.name | ":  " | &PROPERTY.Type;
   &MYFILE.WriteLine(&OutTEXT);
End-For;
```

ItemByName

Syntax

ItemByName(*name*)

Description

The ItemByName method returns the item specified by *name*. *Name* is not case-sensitive.

Parameters

Parameter	Description
name	Specify the name of the PropertyInfo object that you want returned. This parameter takes a string value.

Returns

A reference to a PropertyInfo object or NULL.

Example

```
&PROPERTY = &PROPERTIES.ItemByName("GetHistoryItems");
```

PropertyInfo Properties

This section discusses the PropertyInfo properties in alphabetical order.

Documentation

Description

This property doesn't actually return all the documentation for the class, just a brief description of the class, as a string. This property is read-only.

Name

Description

This property returns the name of the object as a string.

This property is read-only.

Type

Description

This property returns the data type. Valid values are:

- Bool (Boolean).

- Number.
- Float.
- String.
- Variant.
- Blob (Binary large object).
- Any API class name.

This property is read-only.

Usage

Description

This property returns a number that describes in which direction the specified property (or argument) can be passed. The following table describes the valid values.

Value	Description
0	Can be passed into PeopleSoft API.
1	Can be passed out of PeopleSoft API.
2	Can be passed either into or out of PeopleSoft API.

This property is read-only.

Summary of Repository Methods and Properties

This section summarizes the repository methods and properties.

Summary of Repository Methods

Repository objects are instantiated from the Repository property on a session object. This table contains a list of all the Repository objects plus their methods. Methods that can be used by a class are marked with an *X*.

Method	Bindings collection	Namespaces collection	ClassInfo collection	MethodInfo collection	PropertyInfo collection
CreateObject (classname)		**X**			
Generate()	**X**				

Method	Bindings collection	Namespaces collection	ClassInfo collection	MethodInfo collection	PropertyInfo collection
Item(number)	**X**	**X**	**X**	**X**	**X**
ItemByName (name)		**X**	**X**	**X**	**X**

Summary of Repository Properties

Repository collection objects are instantiated from the Repository property on a session object. This table contains a list of all the Repository objects plus their properties (marked with an *RO* in the table). All properties are read-only.

Property	Rep coll/ object	Binding coll/ object	Namespace coll/ object	ClassInfo coll/ object	MethodInfo coll/ object	PropertyInfo coll/ object
Arguments					**RO**	
Bindings	**RO**					
Classes			**RO**			
Count		**RO**	**RO**	**RO**	**RO**	**RO**
Documentation				**RO**	**RO**	**RO**
Generate		**RO**				
Methods				**RO**		
Name		**RO**	**RO**	**RO**	**RO**	**RO**
Namespaces	**RO**					
Properties				**RO**		
Type					**RO**	**RO**
Usage						**RO**

Glossary of PeopleSoft Terms

absence entitlement	This element defines rules for granting paid time off for valid absences, such as sick time, vacation, and maternity leave. An absence entitlement element defines the entitlement amount, frequency, and entitlement period.
absence take	This element defines the conditions that must be met before a payee is entitled to take paid time off.
account	You use an account code to record and summarize financial transactions as expenditures, revenues, assets, or liabilities balances. The use of this delivered PeopleSoft ChartField is typically defined when you implement PeopleSoft General Ledger.
accounting class	In PeopleSoft Enterprise Performance Management, the accounting class defines how a resource is treated for generally accepted accounting practices. The Inventory class indicates whether a resource becomes part of a balance sheet account, such as inventory or fixed assets, while the Non-inventory class indicates that the resource is treated as an expense of the period during which it occurs.
accounting date	The accounting date indicates when a transaction is recognized, as opposed to the date the transaction actually occurred. The accounting date and transaction date can be the same. The accounting date determines the period in the general ledger to which the transaction is to be posted. You can only select an accounting date that falls within an open period in the ledger to which you are posting. The accounting date for an item is normally the invoice date.
accounting entry	A set of related debits and credits. An accounting entry is made up of multiple accounting lines. In most PeopleSoft applications, accounting entries are always balanced (debits equal credits). Accounting entries are created to record accruals, payments, payment cancellations, manual closures, project activities in the general ledger, and so forth, depending on the application.
accounting split	The accounting split method indicates how expenses are allocated or divided among one or more sets of accounting ChartFields.
accumulator	You use an accumulator to store cumulative values of defined items as they are processed. You can accumulate a single value over time or multiple values over time. For example, an accumulator could consist of all voluntary deductions, or all company deductions, enabling you to accumulate amounts. It allows total flexibility for time periods and values accumulated.
action reason	The reason an employee's job or employment information is updated. The action reason is entered in two parts: a personnel action, such as a promotion, termination, or change from one pay group to anotherand a reason for that action. Action reasons are used by PeopleSoft Human Resources, PeopleSoft Benefits Administration, PeopleSoft Stock Administration, and the COBRA Administration feature of the Base Benefits business process.
activity	In PeopleSoft Enterprise Learning Management, an instance of a catalog item delivery methodit may also be called a class. The activity defines such things as meeting times and locations, instructors, reserved equipment and materials, and detailed costs that are associated with the offering, enrollment limits and deadlines, and waitlisting capacities.
allocation rule	In PeopleSoft Enterprise Incentive Management, an expression within compensation plans that enables the system to assign transactions to nodes and participants. During transaction allocation, the allocation engine traverses the compensation structure

	from the current node to the root node, checking each node for plans that contain allocation rules.
alternate account	A feature in PeopleSoft General Ledger that enables you to create a statutory chart of accounts and enter statutory account transactions at the detail transaction level, as required for recording and reporting by some national governments.
application agent	An application agent is an online agent that is loaded into memory with a PeopleSoft page. It detects when a business rule has been triggered and determines the appropriate action.
asset class	An asset group used for reporting purposes. It can be used in conjunction with the asset category to refine asset classification.
attachment	In PeopleSoft Enterprise Learning Management, nonsystem-defined electronic material that supplements a learning resource, such as an equipment items user handbook or the site map of a large facility.
background process	In PeopleSoft, background processes are executed through process-specific COBOL programs and run outside the Windows environment.
benchmark job	In PeopleSoft Workforce Analytics, a benchmark job is a job code for which there is corresponding salary survey data from published, third-party sources.
branch	A tree node that rolls up to nodes above it in the hierarchy, as defined in PeopleSoft Tree Manager.
budgetary account only	An account used by the system only and not by users; this type of account does not accept transactions. You can only budget with this account. Formerly called system-maintained account.
budget check	In commitment control, the processing of source transactions against control budget ledgers, to see if they pass, fail, or pass with a warning.
budget control	In commitment control, budget control ensures that commitments and expenditures don't exceed budgets. It enables you to track transactions against corresponding budgets and terminate a document's cycle if the defined budget conditions are not met. For example, you can prevent a purchase order from being dispatched to a vendor if there are insufficient funds in the related budget to support it.
budget period	The interval of time (such as 12 months or 4 quarters) into which a period is divided for budgetary and reporting purposes. The ChartField allows maximum flexibility to define operational accounting time periods without restriction to only one calendar.
business event	In PeopleSoft Sales Incentive Management, an original business transaction or activity that may justify the creation of a PeopleSoft Enterprise Incentive Management event (a sale, for example).
catalog item	In PeopleSoft Enterprise Learning Management, a specific topic that a learner can study and have tracked. For example, Introduction to Microsoft Word. A catalog item contains general information about the topic and includes a course code, description, categorization, keywords, and delivery methods.
category	In PeopleSoft Enterprise Learning Management, a way to classify catalog items so that users can easily browse and search relevant entries in the learning catalog. Categories can be hierarchical.
ChartField	A field that stores a chart of accounts, resources, and so on, depending on the PeopleSoft application. ChartField values represent individual account numbers, department codes, and so forth.
ChartField balancing	You can require specific ChartFields to match up (balance) on the debit and the credit side of a transaction.

ChartField combination edit The process of editing journal lines for valid ChartField combinations based on user-defined rules.

ChartKey One or more fields that uniquely identify each row in a table. Some tables contain only one field as the key, while others require a combination.

child In PeopleSoft Tree Manager trees, a child is a node or detail on a tree linked to another, higher-level node (referred to as the parent). Child nodes can be rolled up into the parent. A node can be a child and a parent at the same time depending on its location within the tree.

Class ChartField A ChartField value that identifies a unique appropriation budget key when you combine it with a fund, department ID, and program code, as well as a budget period. Formerly called *sub-classification.*

clone In PeopleCode, to make a unique copy. In contrast, to *copy* may mean making a new reference to an object, so if the underlying object is changed, both the copy and the original change.

collection To make a set of documents available for searching in Verity, you must first create at least one collection. A collection is set of directories and files that allow search application users to use the Verity search engine to quickly find and display source documents that match search criteria. A collection is a set of statistics and pointers to the source documents, stored in a proprietary format on a file server. Because a collection can only store information for a single location, PeopleSoft maintains a set of collections (one per language code) for each search index object.

compensation object In PeopleSoft Enterprise Incentive Management, a node within a compensation structure. Compensation objects are the building blocks that make up a compensation structure's hierarchical representation.

compensation structure In PeopleSoft Enterprise Incentive Management, a hierarchical relationship of compensation objects that represents the compensation-related relationship between the objects.

configuration parameter catalog Used to configure an external system with PeopleSoft. For example, a configuration parameter catalog might set up configuration and communication parameters for an external server.

configuration plan In PeopleSoft Enterprise Incentive Management, configuration plans hold allocation information for common variables (not incentive rules) and are attached to a node without a participant. Configuration plans are not processed by transactions.

content reference Content references are pointers to content registered in the portal registry. These are typically either URLs or iScripts. Content references fall into three categories: target content, templates, and template pagelets.

context In PeopleSoft Enterprise Incentive Management, a mechanism that is used to determine the scope of a processing run. PeopleSoft Enterprise Incentive Management uses three types of context: plan, period, and run-level.

corporate account Equivalent to the Account ChartField. Distinguishes between the chart of accounts typically used to record and report financial information for management, stockholders, and the general public, as opposed to a chart of statutory (alternate) accounts required by a regulatory authority for recording and reporting financial information.

cost profile A combination of a receipt cost method, a cost flow, and a deplete cost method. A profile is associated with a cost book and determines how items in that book are valued, as well as how the material movement of the item is valued for the book.

cost row A cost transaction and amount for a set of ChartFields.

data acquisition In PeopleSoft Enterprise Incentive Management, the process during which raw business transactions are acquired from external source systems and fed into the operational data store (ODS).

data elements Data elements, at their simplest level, define a subset of data and the rules by which to group them.

For Workforce Analytics, data elements are rules that tell the system what measures to retrieve about your workforce groups.

data row Contains the entries for each field in a table. To identify each data row uniquely, PeopleSoft applications use a key consisting of one or more fields in the table.

data validation In PeopleSoft Enterprise Incentive Management, a process of validating and cleansing the feed data to resolve conflicts and make the data processable.

DAT file This text file, used with the Verity search engine, contains all of the information from documents that are searchable but not returned in the results list.

delivery method In PeopleSoft Enterprise Learning Management, identifies a learning activitys delivery method type. An activity can have one or more delivery methods.

delivery method type In PeopleSoft Enterprise Learning Management, specifies a method that your organization uses to deliver learning activities, for example, scheduled or self-paced learning.

distribution The process of assigning values to ChartFields. A distribution is a string of ChartField values assigned to items, payments, and budget amounts.

double byte character If youre working with Japanese or other Asian employees, you can enter the employee's name using double-byte characters. The standard double byte character set name format in PeopleSoft applications is: [last name] space [first name].

dynamic tree A tree that takes its detail valuesdynamic detailsdirectly from a table in the database, rather than from a range of values entered by the user.

edit table A table in the database that has its own record definition, such as the Department table. As fields are entered into a PeopleSoft application, they can be validated against an edit table to ensure data integrity throughout the system.

effective date A method of dating information in PeopleSoft applications. You can predate information to add historical data to your system, or postdate information in order to enter it before it actually goes into effect. By using effective dates, you don't delete values; you enter a new value with a current effective date.

EIM job Abbreviation for *Enterprise Incentive Management job.* In PeopleSoft Enterprise Incentive Management, a collection of job steps that corresponds to the steps in an organizations compensation-related business process. An EIM job can be stopped to allow manual changes or corrections to be applied between steps, and then resumed from where it left off, continuing with the next step. A run can also be restarted or rolled back.

EIM ledger Abbreviation for *Enterprise Incentive Management ledger.* In PeopleSoft Enterprise Incentive Management, an object to handle incremental result gathering within the scope of a participant. The ledger captures a result set with all of the appropriate traces to the data origin and to the processing steps of which it is a result.

equipment In PeopleSoft Enterprise Learning Management, resource items that can be assigned to a training facility, to a specific training room, or directly to an activity session. Equipment items are generally items that are used (sometimes for a fee) and returned after the activity is complete.

event Events are predefined points either in the application processor flow or in the program flow. As each point is encountered, the event activates each component, triggering any PeopleCode program associated with that component and that event. Examples of events are FieldChange, SavePreChange, and OnRouteSubscription. In PeopleSoft Human Resources, *event* also refers to incidents that affect benefits eligibility.

event propagation process In PeopleSoft Sales Incentive Management, a process that determines, through logic, the propagation of an original PeopleSoft Enterprise Incentive Management event and creates a derivative (duplicate) of the original event to be processed by other objects. Sales Incentive Management uses this mechanism to implement splits, roll-ups, and so on. Event propagation determines who receives the credit.

external system In PeopleSoft, any system that is not directly compiled with PeopleTools servers.

fact In PeopleSoft applications, facts are numeric data values from fields from a source database as well as an analytic application. A fact can be anything you want to measure your business by, for example, revenue, actual, budget data, or sales numbers. A fact is stored on a fact table.

filter In PeopleSoft applications, a filter creates a subset of information. Filters are used in templates to limit your information from a pick list of attribute values.

generic process type In PeopleSoft Process Scheduler, process types are identified by a generic process type. For example, the generic process type SQR includes all SQR process types, such as SQR process and SQR report.

group Any set of records associated under a single name or variable in order to run calculations in PeopleSoft business processes. In PeopleSoft Time and Labor, for example, employees are placed in groups for time reporting purposes.

homepage Users can personalize the homepage, or the page that first appears when they access the portal.

incentive object In PeopleSoft Enterprise Incentive Management, the incentive-related objects that define and support the PeopleSoft Enterprise Incentive Management calculation process and results, such as plan templates, plans, results data, user interaction objects, and so on.

incentive rule In PeopleSoft Sales Incentive Management, the commands that act on transactions and turn them into compensation. A rule is one part in the process of turning a transaction into compensation.

key One or more fields that uniquely identify each row in a table. Some tables contain only one field as the key, while others require a combination.

learner group In PeopleSoft Enterprise Learning Management, a group of learners within the same learning environment that share the same attributes, such as department or job code.

learning activity See *activity.*

learning history In PeopleSoft Enterprise Learning Management, a self-service repository for all of a learner's completed learning activities.

learning plan In PeopleSoft Enterprise Learning Management, a self-service repository for all of a learner's planned and in-progress learning activities.

ledger mapping You use ledger mapping to relate expense data from general ledger accounts to resource objects. Multiple ledger line items can be mapped to one or more resource IDs. You can also use ledger mapping to map dollar amounts (referred to as *rates*) to business units. You can map the amounts in two different ways: an actual amount that represents actual costs of the accounting period, or a budgeted amount that can be used to calculate the capacity rates as well as budgeted model results. In PeopleSoft Enterprise Warehouse, you can map general ledger accounts to the EW Ledger table.

level	A section of a tree that organizes groups of nodes.
library section	In PeopleSoft Enterprise Incentive Management, a section that is defined in a plan (or template) and that is available for other plans to share. Changes to a library section are reflected in all plans that use it.
linked section	In PeopleSoft Enterprise Incentive Management, a section that is defined in a plan template but appears in a plan. Changes to linked sections propagate to plans using that section.
linked variable	In PeopleSoft Enterprise Incentive Management, a variable that is defined and maintained in a plan template and that also appears in a plan. Changes to linked variables propagate to plans using that variable.
load	The feature that initiates a process to automatically load information into a PeopleSoft applicationfor example, populating the PeopleSoft Benefits database with plan-level election information.
local functionality	In PeopleSoft HRMS, the set of information that is available for a specific country. You can access this information when you click the appropriate country flag in the global window, or when you access it by a local country menu.
location	Locations enable you to indicate the different types of addressesfor a company, for example, one address to receive bills, another for shipping, a third for postal deliveries, and a separate street address. Each address has a different location number. The primary locationindicated by a *1*is the address you use most often and may be different from the main address.
market template	In PeopleSoft Enterprise Incentive Management, additional functionality that is specific to a given market or industry and is built on top of a product category.
material	In PeopleSoft Enterprise Learning Management, a resource item that can be assigned to the sessions of an activity. Material items are generally consumed during the duration of an activity and not returned, and they may have an associated cost.
message definition	An object definition specified in PeopleSoft Application Designer that contains message information for PeopleSoft Application Messaging.
meta-SQL	Meta-SQL constructs expand into platform-specific SQL substrings. They are used in functions that pass SQL strings, such as in SQL objects, the SQLExec function, and PeopleSoft Application Engine programs.
metastring	Metastrings are special expressions included in SQL string literals. The metastrings, prefixed with a percent (%) symbol, are included directly in the string literals. They expand at run time into an appropriate substring for the current database platform.
multibook	Processes in PeopleSoft applications that can create both application entries and general ledgers denominated in more than one currency.
multicurrency	The ability to process transactions in a currency other than the business unit's base currency.
objective	In PeopleSoft Enterprise Learning Management, an individual's learning goal. An example of a learning goal is a competency gap.
override	In PeopleSoft Enterprise Incentive Management, the ability to make a change to a plan that applies to only one plan context.
pagelet	Each block of content on the homepage is called a pagelet. These pagelets display summary information within a small rectangular area on the page. The pagelet provide users with a snapshot of their most relevant PeopleSoft and non-PeopleSoft content.

parent node A tree node linked to lower-level nodes or details that roll up into it. A node can be a parent and a child at the same time, depending on its location within the tree.

participant In PeopleSoft Enterprise Incentive Management, participants are recipients of the incentive compensation calculation process.

participant object Each participant object may be related to one or more compensation objects.

See also *participant object.*

payout In PeopleSoft Enterprise Incentive Management, the resulting incentive plan computation that is provided to payroll.

PeopleCode PeopleCode is a proprietary language, executed by the PeopleSoft application processor. PeopleCode generates results based upon existing data or user actions. By using business interlink objects, external services are available to all PeopleSoft applications wherever PeopleCode can be executed.

PeopleCode event An action that a user takes upon an object, usually a record field, that is referenced within a PeopleSoft page.

PeopleSoft Internet Architecture The fundamental architecture on which PeopleSoft 8 applications are constructed, consisting of an RDBMS, an application server, a Web server, and a browser.

performance measurement In PeopleSoft Enterprise Incentive Management, a variable used to store data (similar to an aggregator, but without a predefined formula) within the scope of an incentive plan. Performance measures are associated with a plan calendar, territory, and participant. Performance measurements are used for quota calculation and reporting.

period context In PeopleSoft Enterprise Incentive Management, because a participant typically uses the same compensation plan for multiple periods, the period context associates a plan context with a specific calendar period and fiscal year. The period context references the associated plan context, thus forming a chain. Each plan context has a corresponding set of period contexts.

per seat cost In PeopleSoft Enterprise Learning Management, the cost per learner, based on the total activity costs divided by either minimum attendees or maximum attendees. Organizations use this cost to price PeopleSoft Enterprise Learning Management activities.

plan In PeopleSoft Sales Incentive Management, a collection of allocation rules, variables, steps, sections, and incentive rules that instruct the PeopleSoft Enterprise Incentive Management engine in how to process transactions.

plan context In PeopleSoft Enterprise Incentive Management, correlates a participant with the compensation plan and node to which the participant is assigned, enabling the PeopleSoft Enterprise Incentive Management system to find anything that is associated with the node and that is required to perform compensation processing. Each participant, node, and plan combination represents a unique plan contextif three participants are on a compensation structure, each has a different plan context. Configuration plans are identified by plan contexts and are associated with the participants that refer to them.

plan section In PeopleSoft Enterprise Incentive Management, a segment of a plan that handles a specific type of event processing.

plan template In PeopleSoft Enterprise Incentive Management, the base from which a plan is created. A plan template contains common sections and variables that are inherited by all plans that are created from the template. A template may contain steps and sections that are not visible in the plan definition.

portal registry In PeopleSoft applications, the portal registry is a tree-like structure in which content references are organized, classified, and registered. It is a central repository that

	defines both the structure and content of a portal through a hierarchical, tree-like structure of folders useful for organizing and securing content references.
private view	A user-defined view that is available only to the user who created it.
process	See *Batch Processes.*
process definition	Process definitions define each run request.
process instance	A unique number that identifies each process request. This value is automatically incremented and assigned to each requested process when the process is submitted to run.
process job	You can link process definitions into a job request and process each request serially or in parallel. You can also initiate subsequent processes based on the return code from each prior request.
process request	A single run request, such as an SQR, a COBOL program, or a Crystal report that you run through PeopleSoft Process Scheduler.
process run control	A PeopleTools variable used to retain PeopleSoft Process Scheduler values needed at runtime for all requests that reference a run control ID. Do not confuse these with application run controls, which may be defined with the same run control ID, but only contain information specific to a given application process request.
product category	In PeopleSoft Enterprise Incentive Management, indicates an application in the Enterprise Incentive Management suite of products. Each transaction in the PeopleSoft Enterprise Incentive Management system is associated with a product category.
publishing	In PeopleSoft Enterprise Incentive Management, a stage in processing that makes incentive-related results available to participants.
record definition	A logical grouping of data elements.
record field	A field within a record definition.
record group	A set of logically and functionally related control tables and views. Record groups help enable TableSet sharing, which eliminates redundant data entry. Record groups ensure that TableSet sharing is applied consistently across all related tables and views.
record input VAT flag	Abbreviation for *record input value-added tax flag.* Within PeopleSoft Purchasing, Payables, and General Ledger, this flag indicates that you are recording input VAT on the transaction. This flag, in conjunction with the record output VAT flag, is used to determine the accounting entries created for a transaction and to determine how a transaction is reported on the VAT return. For all cases within Purchasing and Payables where VAT information is tracked on a transaction, this flag is set to Yes. This flag is not used in PeopleSoft Order Management, Billing, or Receivables, where it is assumed that you are always recording only output VAT, or in PeopleSoft Expenses, where it is assumed that you are always recording only input VAT.
record output VAT flag	Abbreviation for *record output value-added tax flag.* See *record input VAT flag.*
reference data	In PeopleSoft Sales Incentive Management, system objects that represent the sales organization, such as territories, participants, products, customers, channels, and so on.
reference object	In PeopleSoft Enterprise Incentive Management, this dimension-type object further defines the business. Reference objects can have their own hierarchy (for example, product tree, customer tree, industry tree, and geography tree).
reference transaction	In commitment control, a reference transaction is a source transaction that is referenced by a higher-level (and usually later) source transaction, in order to

	automatically reverse all or part of the referenced transaction's budget-checked amount. This avoids duplicate postings during the sequential entry of the transaction at different commitment levels. For example, the amount of an encumbrance transaction (such as a purchase order) will, when checked and recorded against a budget, cause the system to concurrently reference and relieve all or part of the amount of a corresponding pre-encumbrance transaction, such as a purchase requisition.
relationship object	In PeopleSoft Enterprise Incentive Management, these objects further define a compensation structure to resolve transactions by establishing associations between compensation objects and business objects.
results management process	In PeopleSoft Sales Incentive Management, the process during which compensation administrators may review processing results, manually change transactions, process draws, update and review payouts, process approvals, and accumulate and push payments to the EIM ledger.
role user	A PeopleSoft Workflow user. A person's role user ID serves much the same purpose as a user ID does in other parts of the system. PeopleSoft Workflow uses role user IDs to determine how to route worklist items to users (through an email address, for example) and to track the roles that users play in the workflow. Role users do not need PeopleSoft user IDs.
role	Describes how people fit into PeopleSoft Workflow. A role is a class of users who perform the same type of work, such as clerks or managers. Your business rules typically specify what user role needs to do an activity.
roll up	In a tree, to roll up is to total sums based on the information hierarchy.
routing	Connects activities in PeopleSoft Workflow. Routings specify where the information goes and what form it takesemail message, electronic form, or worklist entry.
run control	A run control is a type of online page that is used to begin a process, such as the batch processing of a payroll run. Run control pages generally start a program that manipulates data.
run control ID	A unique ID to associate each user with his or her own run control table entries.
run-level context	In PeopleSoft Enterprise Incentive Management, associates a particular run (and batch ID) with a period context and plan context. Every plan context that participates in a run has a separate run-level context. Because a run cannot span periods, only one run-level context is associated with each plan context.
search query	You use this set of objects to pass a query string and operators to the search engine. The search index returns a set of matching results with keys to the source documents.
section	In PeopleSoft Enterprise Incentive Management, a collection of incentive rules that operate on transactions of a specific type. Sections enable plans to be segmented to process logical events in different sections.
security event	In commitment control, security events trigger security authorization checking, such as budget entries, transfers, and adjustments; exception overrides and notifications; and inquiries.
self-service application	Self-service refers to PeopleSoft applications that are accessed by end users with a browser.
session	In PeopleSoft Enterprise Learning Management, a single meeting day of an activity (that is, the period of time between start and finish times within a day). The session stores the specific date, location, meeting time, and instructor. Sessions are used for scheduled training.
session template	In PeopleSoft Enterprise Learning Management, enables you to set up common activity characteristics that may be reused while scheduling a PeopleSoft Enterprise

	Learning Management activitycharacteristics such as days of the week, start and end times, facility and room assignments, instructors, and equipment. A session pattern template can be attached to an activity that is being scheduled. Attaching a template to an activity causes all of the default template information to populate the activity session pattern.
setup relationship	In PeopleSoft Enterprise Incentive Management, a relationship object type that associates a configuration plan with any structure node.
sibling	A tree node at the same level as another node, where both roll up into the same parent. A node can be a sibling, parent, and child all at the same time, depending on its location in the tree.
single signon	With single signon, users can, after being authenticated by a PeopleSoft application server, access a second PeopleSoft application server without entering a user ID or password.
source transaction	In commitment control, any transaction generated in a PeopleSoft or third-party application that is integrated with commitment control and which can be checked against commitment control budgets. For example, a pre-encumbrance, encumbrance, expenditure, recognized revenue, or collected revenue transaction.
SpeedChart	A user-defined shorthand key that designates several ChartKeys to be used for voucher entry. Percentages can optionally be related to each ChartKey in a SpeedChart definition.
SpeedType	A code representing a combination of ChartField values. SpeedTypes simplify the entry of ChartFields commonly used together.
SQR	See *Structured Query Report (SQR)*.
statutory account	Account required by a regulatory authority for recording and reporting financial results. In PeopleSoft, this is equivalent to the Alternate Account (ALTACCT) ChartField.
step	In PeopleSoft Sales Incentive Management, a collection of sections in a plan. Each step corresponds to a step in the job run.
Structured Query Report (SQR)	A type of printed or displayed report generated from data extracted from a PeopleSoft SQL-based relational database. PeopleSoft applications provide a variety of standard SQRs that summarize table information and data. You can use these reports as is, customize them, or create your own.
Summary ChartField	You use summary ChartFields to create summary ledgers that roll up detail amounts based on specific detail values or on selected tree nodes. When detail values are summarized using tree nodes, summary ChartFields must be used in the summary ledger data record to accommodate the maximum length of a node name (20 characters).
summary ledger	An accounting feature used primarily in allocations, inquiries, and PS/nVision reporting to store combined account balances from detail ledgers. Summary ledgers increase speed and efficiency of reporting by eliminating the need to summarize detail ledger balances each time a report is requested. Instead, detail balances are summarized in a background process according to user-specified criteria and stored on summary ledgers. The summary ledgers are then accessed directly for reporting.
summary tree	A tree used to roll up accounts for each type of report in summary ledgers. Summary trees enable you to define trees on trees. In a summary tree, the detail values are really nodes on a detail tree or another summary tree (known as the *basis* tree). A summary tree structure specifies the details on which the summary trees are to be built.

table The underlying PeopleSoft data format, in which data is stored by columns (fields) and rows (records, or instances).

TableSet sharing Specifies control table data for each business unit so that redundancy is eliminated.

target currency The value of the entry currency or currencies converted to a single currency for budget viewing and inquiry purposes.

template A template is HTML code associated with a Web page. It defines the layout of the page and also where to get HTML for each part of the page. In PeopleSoft, you use templates to build a page by combining HTML from a number of sources. For a PeopleSoft portal, all templates must be registered in the portal registry, and each content reference must be assigned a template.

territory In PeopleSoft Sales Incentive Management, hierarchical relationships of business objects, including regions, products, customers, industries, and participants.

TimeSpan A relative period, such as year-to-date or current period, that can be used in various PeopleSoft General Ledger functions and reports when a rolling time frame, rather than a specific date, is required. TimeSpans can also be used with flexible formulas in PeopleSoft Projects.

transaction allocation In PeopleSoft Enterprise Incentive Management, the process of identifying the owner of a transaction. When a raw transaction from a batch is allocated to a plan context, the transaction is duplicated in the PeopleSoft Enterprise Incentive Management transaction tables.

transaction loading process In PeopleSoft Enterprise Incentive Management, the process during which transactions are loaded into Sales Incentive Management. During loading, the source currency is converted to the business unit currency while retaining the source currency code. At the completion of this stage, the transaction is in the first state.

transaction state In PeopleSoft Enterprise Incentive Management, a value assigned by an incentive rule to a transaction. Transaction states enable sections to process only transactions that are at a specific stage in system processing. After being successfully processed, transactions may be promoted to the next transaction state and picked up by a different section for further processing.

transaction type In PeopleSoft Enterprise Incentive Management, a way to categorize transactions to identify specific transaction types (for example, shipment, order, opportunity, and so on). Plan sections process only one type of transaction type. Transaction types can be defined based on a company's specific processes model.

Translate table A system edit table that stores codes and translate values for the miscellaneous fields in the database that do not warrant individual edit tables of their own.

tree The graphical hierarchy in PeopleSoft systems that displays the relationship between all accounting units (for example, corporate divisions, projects, reporting groups, account numbers) and determines roll-up hierarchies.

unclaimed transaction In PeopleSoft Enterprise Incentive Management, a transaction that is not claimed by a node or participant after the allocation process has completed, usually due to missing or incomplete data. Unclaimed transactions may be manually assigned to the appropriate node or participant by a compensation administrator.

uniform resource locator (URL) In PeopleSoft, the term URL refers to the entire query string. The following is an example of a URL: http://serverx/InternetClient /InternetClientServlet?ICType=Script&ICScriptProgramName=WEBLIB_BEN_ 401k.PAGES.FieldFormula.iScript_Home401k

universal navigation header Every PeopleSoft portal includes the universal navigation header, intended to appear at the top of every page as long as the user is signed on to the portal. In addition to

	providing access to the standard navigation buttons (like Home, Favorites, and signoff) the universal navigation header can also display a welcome message for each user.
URL	See *uniform resource locator (URL).*
user interaction object	In PeopleSoft Sales Incentive Management, used to define the reporting components and reports that a participant can access in his or her context. All Sales Incentive Management user interface objects and reports are registered as user interaction objects. User interaction objects can be linked to a compensation structure node through a compensation relationship object (individually or as groups).
variable	In PeopleSoft Sales Incentive Management, the intermediate results of calculations. Variables hold the calculation results and are then inputs to other calculations. Variables can be plan variables that persist beyond the run of an engine or local variables that exist only during the processing of a section.
warehouse	A PeopleSoft data warehouse that consists of predefined ETL maps, data warehouse tools, and DataMart definitions.
worksheet	A way of presenting data through a PeopleSoft Business Analysis Modeler interface that enables users to do in-depth analysis using pivoting tables, charts, notes, and history information.
workflow	The background process that creates a list of administrative actions based on selection criteria and specifies the procedure associated with each action.
worklist	The automated to-do list that PeopleSoft Workflow creates. From the worklist, you can directly access the pages you need to perform the next action, and then return to the worklist for another item.
zero-rated VAT	Abbreviation for *zero-rated value-added tax.* A VAT transaction with a VAT code that has a tax percent of zero. Used to track taxable VAT activity where no actual VAT amount is charged.

Index